

IBM Storage Scale
5.2.1

Administration Guide



Note

Before using this information and the product it supports, read the information in “[Notices](#)” on page [1003](#).

This edition applies to Version 5 release 2 modification 1 of the following products, and to all subsequent releases and modifications until otherwise indicated in new editions:

- IBM Storage Scale Data Management Edition ordered through Passport Advantage® (product number 5737-F34)
- IBM Storage Scale Data Access Edition ordered through Passport Advantage (product number 5737-I39)
- IBM Storage Scale Erasure Code Edition ordered through Passport Advantage (product number 5737-J34)
- IBM Storage Scale Data Management Edition ordered through AAS (product numbers 5641-DM1, DM3, DM5)
- IBM Storage Scale Data Access Edition ordered through AAS (product numbers 5641-DA1, DA3, DA5)
- IBM Storage Scale Data Management Edition for IBM® ESS (product number 5765-DME)
- IBM Storage Scale Data Access Edition for IBM ESS (product number 5765-DAE)
- IBM Storage Scale Backup ordered through Passport Advantage® (product number 5900-AXJ)
- IBM Storage Scale Backup ordered through AAS (product numbers 5641-BU1, BU3, BU5)
- IBM Storage Scale Backup for IBM® Storage Scale System (product number 5765-BU1)

Significant changes or additions to the text and illustrations are indicated by a vertical line (|) to the left of the change.

IBM welcomes your comments; see the topic “[How to send your comments](#)” on page [xlvi](#). When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2015, 2024.

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

Contents

Tables.....	xvii
About this information.....	xxi
Prerequisite and related information.....	xl
Conventions used in this information.....	xli
How to send your comments.....	xlii
Summary of changes.....	xliii
Chapter 1. Configuring the GPFS cluster.....	1
Creating your GPFS cluster.....	1
Displaying cluster configuration information.....	2
Adding nodes to a GPFS cluster.....	4
Deleting nodes from a GPFS cluster.....	5
Changing the GPFS cluster configuration data.....	8
Security mode.....	22
Setting the security mode for internode communications in a cluster.....	23
Minimum release level of a cluster.....	24
Running IBM Storage Scale commands without remote root login.....	26
Configuring sudo.....	28
Configuring the cluster to use sudo wrapper scripts.....	29
Configuring IBM Storage Scale GUI to use sudo wrapper.....	30
Configuring a cluster to stop using sudo wrapper scripts.....	30
Root-level processes that call administration commands directly.....	30
Cluster quorum with quorum nodes.....	30
Cluster quorum with quorum nodes and tiebreaker disks.....	31
Displaying and changing the file system manager node.....	33
Starting and stopping GPFS.....	34
Starting or stopping GPFS daemon on a node by using GUI.....	36
Shutting down an IBM Storage Scale cluster.....	36
Configuring cluster configuration repository.....	38
Enabling CCR.....	39
CCR directory structure and recommendations for configuring CCR.....	39
Disaster recovery scenarios for CCR.....	40
Limitations of CCR.....	41
Chapter 2. Configuring GPUDirect Storage for IBM Storage Scale.....	45
Chapter 3. Configuring the CES and protocol configuration.....	49
Configuring Cluster Export Services	49
Setting up Cluster Export Services shared root file system.....	49
Configuring Cluster Export Services nodes.....	51
Configuring CES protocol service IP addresses.....	52
CES IP aliasing to network adapters on protocol nodes.....	53
Deploying Cluster Export Services packages on existing IBM Storage Scale nodes.....	58
Verifying the final CES configurations.....	59
Creating and configuring file systems and filesets for exports.....	60
Configuring with the installation toolkit.....	60
Deleting a Cluster Export Services node from an IBM Storage Scale cluster.....	61

Setting up Cluster Export Services groups in an IBM Storage Scale cluster.....	61
Setting up self-signed SSL/TLS certificates for secure communication between the S3 client and the S3 service	62
Chapter 4. Configuring and tuning your system for GPFS.....	65
General system configuration and tuning considerations.....	65
Clock synchronization.....	65
GPFS administration security.....	66
Cache usage.....	66
Access patterns.....	69
Aggregate network interfaces.....	69
Swap space.....	70
Linux configuration and tuning considerations.....	70
updatedb considerations.....	71
Memory considerations.....	71
GPFS helper threads.....	71
Communications I/O.....	71
Disk I/O.....	72
AIX configuration and tuning considerations.....	72
GPFS use with Oracle.....	72
Chapter 5. Parameters for performance tuning and optimization.....	75
Recommendations for tuning <code>maxTcpConnsPerNodeConn</code> parameter.....	77
Tuning parameters change history.....	79
Chapter 6. Ensuring high availability of the GUI service.....	85
Chapter 7. Configuring and tuning your system for Cloud services.....	87
Configuration command execution matrix.....	87
Designating the Cloud services nodes.....	88
Starting up the Cloud services software.....	89
Managing a cloud storage account.....	90
Amazon S3.....	90
Swift3 account.....	91
IBM Cloud Object Storage.....	91
Microsoft Azure.....	92
Defining cloud storage access points (CSAP).....	92
Creating Cloud services.....	94
Configuring Cloud services with SKLM (optional).....	95
Binding your file system or fileset to the Cloud service by creating a container pair set.....	96
Backing up the Cloud services database to the cloud.....	99
Backing up the Cloud services configuration.....	99
Configuring the maintenance windows.....	100
Enabling a policy for Cloud data sharing export service.....	102
Tuning Cloud services parameters.....	103
Integrating Cloud services metrics with the performance monitoring tool.....	106
GPFS-based configuration.....	106
File-based configuration.....	107
Setting up Transparent cloud tiering service on a remotely mounted client.....	109
Deploying WORM solutions.....	110
Creating immutable filesets and files	111
Setting up Transparent cloud tiering for WORM solutions.....	112
Chapter 8. Configuring IBM Power Systems for IBM Storage Scale.....	121
Tuning the operating system.....	121
Logical partitioning (LPAR) hardware allocations for NUMA-based Power servers.....	122
Running Dynamic Platform Optimizer (DPO) to optimize an LPAR.....	123

Configuring INT_LOG_MAX_PAYLOAD_SIZE Parameter.....	123
Chapter 9. Configuring file audit logging.....	125
Enabling file audit logging on a file system.....	125
Disabling file audit logging on a file system.....	125
Enabling or skipping filesets with file audit logging.....	126
Actions that the mmaudit command takes to enable file audit logging.....	126
Actions that the mmaudit command takes to disable file audit logging.....	127
Enabling and disabling file audit logging using the GUI.....	128
Viewing file systems that have file audit logging enabled with the GUI.....	128
Enabling file audit logging on an owning cluster for a file system that is remotely mounted.....	128
Chapter 10. Configuring clustered watch folder.....	129
Enabling a clustered watch.....	129
Disabling a clustered watch.....	129
Configuration of an external Kafka sink in the IBM Storage Scale cluster.....	129
Actions that the mmwatch command takes to enable a clustered watch.....	129
Actions that the mmwatch command takes to disable a clustered watch.....	130
Chapter 11. Configuring the cloudbit.....	131
Configuring your AWS cloud account.....	131
Configuring your GCP cloud account.....	131
Configuring your Azure cloud account.....	132
Chapter 12. Configuring Active File Management.....	133
Configuration parameters for AFM, AFM-DR, and AFM to cloud object storage.....	133
Configuration changes in an existing AFM relationship.....	145
Adding gateway nodes to the cache cluster.....	145
The NFS server at the home cluster.....	145
Enabling AFM Network File System version 4.....	146
Mapping IDs with the AFM Network File System version 4.....	146
Chapter 13. Configuring AFM-based DR.....	149
Changing configuration in an existing AFM DR relationship.....	149
Changing NFS server on secondary.....	149
Changing gateway nodes on primary.....	149
Chapter 14. Configuring AFM to cloud object storage.....	151
Configuring an AFM to cloud object storage filesset with Microsoft Azure Blob.....	156
Configuring AFM to cloud object storage for Azure Blob storage by using MinIO as S3 gateway.....	159
Configuring AFM to cloud object storage filesset by using use-keys and STS token.....	159
Configuring AFM to cloud object storage to use Google cloud storage.....	160
Configuring the replication at the file system level by using the manual updates mode of the AFM to cloud object storage.....	160
Configuring a new file system for the replication by using the manual updates mode of the AFM to cloud object storage.....	160
Configuring an existing file system for the replication by using the manual updates mode of the AFM to cloud object storage.....	164
Configuring the multi-site replication of AFM to cloud object storage	167
Chapter 15. Tuning for Kernel NFS backend on AFM and AFM DR.....	169
Tuning the gateway node on the NFS client.....	169
Tuning on both the NFS client (gateway) and the NFS server (the home/secondary cluster).....	169
Tuning the NFS server on the home/secondary cluster or the NFS server.....	170
Chapter 16. Configuring call home.....	173

Configuring call home to enable manual and automated data upload.....	173
Configuring the call home groups manually.....	174
Configuring the call home groups automatically.....	174
Configuring call home using GUI.....	175
Call home configuration examples.....	176
Use cases for detecting system changes by using the mmcallhome command.....	179
Chapter 17. Integrating IBM Storage Scale Cinder driver with Red Hat	
OpenStack Platform 16.1.....	183
Configuring IBM Storage Scale cluster to enable Cinder driver with RHOSP.....	183
Deploying IBM Storage Scale Cinder backend configuration in RHOSP.....	184
Limitations of integrating IBM Storage Scale Cinder driver with RHOSP.....	185
Triple-O heat template environment parameters.....	185
Sample IBM Storage Scale Cinder configuration YAML file.....	186
Chapter 18. Configuring Multi-Rail over TCP (MROT).....	189
Chapter 19. Dynamic pagepool configuration.....	195
Chapter 20. Configuring shared memory communications direct.....	197
Verifying SMC-D requirements on each node.....	198
Chapter 21. Performing GPFS administration tasks.....	199
Requirements for administering a GPFS file system.....	199
The adminMode configuration attribute.....	200
Common GPFS command principles.....	201
Specifying nodes as input to GPFS commands.....	201
Stanza files.....	202
Listing active IBM Storage Scale commands.....	203
Determining how long mmrestripesfs takes to complete.....	204
Chapter 22. Performing parallel copy with mmxcp command.....	205
Chapter 23. Managing shared memory communications direct.....	209
Chapter 24. Protecting file data: IBM Storage Scale safeguarded copy.....	211
Chapter 25. Verifying network operation with the mmnetverify command.....	215
Chapter 26. Managing file systems.....	217
Mounting a file system.....	217
Mounting a file system on multiple nodes.....	218
Mount options specific to IBM Storage Scale.....	218
Mounting a file system through GUI	219
Changing a file system mount point on protocol nodes.....	220
Unmounting a file system.....	221
Unmounting a file system on multiple nodes.....	221
Unmounting a file system through GUI	222
Deleting a file system.....	222
Determining which nodes have a file system mounted.....	223
Checking and repairing a file system.....	223
Dynamic validation of descriptors on disk.....	225
File system maintenance mode	225
Listing file system attributes.....	228
Modifying file system attributes.....	229

Querying and changing file replication attributes.....	229
Querying file replication.....	230
Changing file replication attributes.....	230
Using Direct I/O on a file in a GPFS file system.....	231
File compression.....	231
Setting the Quality of Service for I/O operations.....	237
Restriping a GPFS file system.....	240
Querying file system space.....	241
Querying and reducing file system fragmentation.....	242
Querying file system fragmentation.....	242
Reducing file system fragmentation.....	243
Protecting data in a file system using backup.....	244
Protecting data in a file system using the mmbackup command.....	244
Backing up a file system using the GPFS policy engine.....	249
Backing up file system configuration information.....	250
Using APIs to develop backup applications.....	250
Scale Out Backup and Restore (SOBAR).....	251
Scheduling backups using IBM Storage Protect scheduler.....	252
Configuration reference for using IBM Storage Protect with IBM Storage Scale.....	252
Options in the IBM Storage Protect configuration file dsm.sys.....	253
Options in the IBM Storage Protect configuration file dsm.opt.....	255
Base IBM Storage Protect client configuration files for IBM Storage Scale usage.....	257
Restoring a subset of files or directories from a local file system snapshot.....	258
Restoring a subset of files or directories from a local filesset snapshot.....	259
Restoring a subset of files or directories from local snapshots using the sample script.....	260
Creating and managing file systems by using GUI.....	261

Chapter 27. File system format changes between versions of IBM Storage Scale. 267

Chapter 28. Managing disks.....	275
Displaying disks in a GPFS cluster.....	275
Adding disks to a file system.....	276
Deleting disks from a file system.....	276
Replacing disks in a GPFS file system.....	278
Additional considerations for managing disks.....	279
Displaying GPFS disk states.....	280
Disk availability.....	280
Disk status.....	280
Changing GPFS disk states and parameters.....	281
Changing your NSD configuration.....	283
Changing NSD server usage and failback.....	284
NSD servers: Periodic checks for I/O problems.....	284
Enabling and disabling Persistent Reserve.....	284

Chapter 29. Managing protocol services.....

Configuring and enabling SMB, S3, and NFS protocol services.....	287
Support of vfs_fruit for the SMB protocol.....	288
Configuring and enabling the Swift Object protocol service.....	290
Disabling protocol services.....	291

Chapter 30. Managing protocol user authentication.....

Setting up authentication servers to configure protocol user access.....	293
Integrating with AD server.....	294
Integrating with LDAP server.....	295
Configuring authentication and ID mapping for file access.....	300
Prerequisites.....	300
Configuring file authentication by using CLI.....	303

Configuring file authentication by using GUI.....	323
Configuring authentication for Swift Object access.....	325
Managing user-defined authentication.....	326
Listing the authentication configuration.....	330
Verifying the authentication services configured in the system.....	331
Modifying the authentication method	332
Deleting the authentication and the ID-mapping configuration.....	333
Authentication limitations.....	335
Chapter 31. Managing protocol data exports.....	339
Managing SMB shares.....	339
Creating SMB share.....	339
Changing SMB share configuration.....	340
Creating SMB share ACLs.....	341
Removing SMB shares.....	341
Listing SMB shares.....	341
Managing SMB shares using MMC.....	342
Managing NFS exports.....	351
Creating NFS exports.....	351
Changing NFS export configuration.....	352
Removing NFS exports.....	353
Listing NFS exports.....	353
GUI navigation for NFS exports.....	353
Making bulk changes to NFS exports.....	353
Multiprotocol exports.....	356
Multiprotocol export considerations.....	356
Chapter 32. Managing S3 protocol.....	359
Managing S3 accounts and buckets.....	359
Managing S3 accounts.....	359
Managing S3 buckets using AWS CLI.....	359
Managing S3 buckets using the mms3 command.....	360
S3 objects API.....	361
S3 buckets API.....	362
Backing up the S3 configuration data.....	362
Restoring the S3 configuration data.....	363
Chapter 33. Managing Swift Object storage.....	367
Understanding and managing Object services.....	367
Understanding the mapping of OpenStack commands to IBM Storage Scale administrator commands.....	369
Changing Object configuration values.....	370
How to change the object base configuration to enable S3 API.....	370
Configuring OpenStack EC2 credentials.....	371
How to manage the OpenStack S3 API.....	372
Managing object capabilities.....	373
Managing object versioning	374
Enabling object versioning.....	374
Disabling object versioning.....	375
Creating a version of an object: Example.....	375
Mapping of storage policies to filesets.....	377
Administering storage policies for Swift Object storage.....	377
Creating storage policy for object compression.....	378
Creating storage policy for object encryption.....	379
Adding a region in a multi-region Swift Object deployment.....	380
Administering a multi-region object deployment environment.....	381
Unified file and Swift Object access in IBM Storage Scale	382

Enabling object access to existing filesets.....	382
Identity management modes for unified file and Swift Object access.....	384
Authentication in unified file and object access.....	389
Validating shared authentication ID mapping.....	390
The objectizer process.....	391
File path in unified file and Swift Object access.....	392
Administering unified file and object access.....	394
In-place analytics using unified file and object access.....	407
Limitations of unified file and object access.....	408
Constraints applicable to unified file and object access.....	409
Data ingestion examples.....	410
curl commands for unified file and object access related user tasks.....	411
Configuration files for IBM Storage Scale for object storage.....	412
Backing up and restoring object storage.....	416
Backing up the object storage.....	416
Restoring the object storage.....	418
Configuration of object for isolated node and network groups.....	420
Enabling the object heatmap policy.....	422
Chapter 34. Managing GPFS quotas.....	425
Enabling and disabling GPFS quota management.....	425
Default quotas.....	427
Implications of quotas for different protocols.....	430
Explicitly establishing and changing quotas.....	431
Setting quotas for users on a per-project basis.....	433
Checking quotas.....	436
Listing quotas.....	438
Activating quota limit checking.....	440
Deactivating quota limit checking.....	441
Changing the scope of quota limit checking.....	443
Creating file system quota reports.....	444
Restoring quota files.....	445
Managing quota by using GUI.....	447
Chapter 35. Managing GUI users.....	451
Create GUI users and assign user permissions.....	453
Defining a password policy for GUI users.....	454
Changing or expiring password of GUI user.....	455
Configuring external authentication for GUI users.....	455
Configuring multi-factor authentication for GUI users.....	458
Configuring GUI details in IBM Security Verify for multi-factor authentication.....	459
Chapter 36. Managing GPFS access control lists.....	461
Traditional GPFS ACL administration.....	461
Setting traditional GPFS access control lists.....	462
Displaying traditional GPFS access control lists.....	463
Applying an existing traditional GPFS access control list.....	464
Changing traditional GPFS access control lists.....	465
Deleting traditional GPFS access control lists.....	465
NFS V4 ACL administration.....	465
NFS V4 ACL Syntax.....	466
NFS V4 ACL translation.....	469
Setting NFS V4 access control lists.....	470
Displaying NFS V4 access control lists.....	471
Applying an existing NFS V4 access control list.....	471
Changing NFS V4 access control lists.....	472
Deleting NFS V4 access control lists.....	472

Considerations when using GPFS with NFS V4 ACLs.....	472
Exceptions and limitations to NFS V4 ACLs support.....	473
Authorizing protocol users.....	474
Authorizing file protocol users.....	475
Authorizing object users.....	489
Authorization limitations.....	496
Chapter 37. Native NFS and GPFS.....	499
Exporting a GPFS file system using NFS.....	499
Export considerations.....	500
NFS usage of GPFS cache.....	502
Synchronous writing using NFS.....	502
Unmounting a file system after NFS export.....	502
NFS automount considerations.....	503
Clustered NFS and GPFS on Linux.....	503
Chapter 38. Accessing a remote GPFS file system.....	505
Remote user access to a GPFS file system.....	507
Using NFS/SMB protocol over remote cluster mounts.....	508
Configuring protocols on a separate cluster.....	509
Managing multi-cluster protocol environments.....	510
Upgrading multi-cluster environments.....	510
Limitations of protocols on remotely mounted file systems.....	511
S3 protocol over remote cluster mounts.....	512
Mounting a remote GPFS file system.....	512
Fileset access control for remote clusters.....	515
Managing remote access to a GPFS file system.....	516
Attaching direct storage on IBM Z.....	516
Using remote access with multiple network definitions.....	518
Using multiple security levels for remote access.....	520
Changing security keys with remote access.....	521
NIST compliance.....	522
Important information about remote access.....	523
Chapter 39. Information lifecycle management for IBM Storage Scale.....	525
Storage pools.....	525
Internal storage pools.....	526
External storage pools.....	530
Policies for automating file management.....	531
Overview of policies.....	531
Policy rules.....	533
The mmapplypolicy command and policy rules.....	553
Policy rules: Examples and tips.....	557
Managing policies.....	563
Working with external storage pools.....	570
Backup and restore with storage pools.....	576
ILM for snapshots.....	578
User storage pools.....	579
File heat: Tracking file access temperature.....	579
Filesets.....	582
Fileset namespace.....	582
Filesets and quotas.....	583
Filesets and storage pools.....	584
Filesets and global snapshots.....	585
Fileset-level snapshots.....	586
Filesets and backup.....	586
Managing filesets.....	588

Immutability and appendOnly features.....	594
Creating and applying ILM policy by using GUI	598
Modifying active ILM policy by using GUI	599
Chapter 40. Creating and maintaining snapshots of file systems.....	601
Creating a snapshot.....	601
Creating a snapshot by using GUI.....	603
Listing snapshots.....	603
Restoring a file system from a snapshot.....	604
Reading a snapshot with the policy engine.....	605
Linking to a snapshot.....	606
Deleting a snapshot.....	607
Managing snapshots using IBM Storage Scale GUI.....	608
Chapter 41. Creating and managing file clones.....	613
Creating file clones.....	613
Listing file clones.....	614
Deleting file clones.....	615
Splitting file clones from clone parents.....	615
File clones and disk space management.....	615
File clones and snapshots.....	615
File clones and policy files.....	616
Chapter 42. Scale Out Backup and Restore (SOBAR).....	617
Backup procedure with SOBAR.....	617
Restore procedure with SOBAR.....	619
Chapter 43. Data Mirroring and Replication.....	623
General considerations for using storage replication with GPFS.....	624
Data integrity and the use of consistency groups.....	624
Handling multiple versions of IBM Storage Scale data.....	624
Continuous Replication of IBM Storage Scale data.....	625
Synchronous mirroring with GPFS replication.....	625
Synchronous mirroring utilizing storage based replication.....	634
Point-in-time copy of IBM Storage Scale data.....	642
Chapter 44. Implementing a clustered NFS environment on Linux.....	645
NFS monitoring.....	645
NFS failover.....	645
NFS locking and load balancing.....	646
CNFS network setup.....	646
CNFS setup.....	647
CNFS administration.....	648
Chapter 45. Implementing Cluster Export Services.....	651
CES features.....	651
CES cluster setup.....	651
Suspending or resuming CES nodes by using GUI.....	652
CES network configuration.....	652
CES address failover and distribution policies.....	654
CES protocol management.....	655
CES management and administration.....	655
CES NFS support.....	656
CES SMB support.....	658
CES OBJ support.....	659
CES HDFS support.....	662

Migration of CNFS clusters to CES clusters.....	662
Chapter 46. Identity management on Windows / RFC 2307 attributes.....	665
Auto-generated ID mappings.....	665
Configuring ID mappings in Active Directory Users and Computers for Windows Server 2016 (and subsequent) versions.....	666
Installing Windows IDMU.....	669
Configuring ID mappings in IDMU.....	670
Chapter 47. Protocols cluster disaster recovery.....	673
Protocols cluster disaster recovery limitations and prerequisites.....	673
Example setup for protocols disaster recovery.....	674
Setting up gateway nodes to ensure cluster communication during failover.....	674
Protocols and cluster configuration data required for disaster recovery.....	675
Swift Object data required for protocols cluster DR.....	675
SMB data required for protocols cluster DR.....	676
NFS data required for protocols cluster DR.....	677
Authentication related data required for protocols cluster DR.....	679
CES data required for protocols cluster DR.....	680
Chapter 48. File Placement Optimizer.....	683
Distributing data across a cluster	687
FPO pool file placement and AFM.....	687
Configuring FPO.....	688
Configuring IBM Storage Scale Clusters.....	688
Basic Configuration Recommendations.....	694
Configuration and tuning of Hadoop workloads.....	705
Configuration and tuning of database workloads.....	706
Configuring and tuning SparkWorkloads.....	706
Ingesting data into IBM Storage Scale clusters.....	707
Exporting data out of IBM Storage Scale clusters.....	707
Upgrading FPO.....	708
Monitoring and administering IBM Storage Scale FPO clusters.....	710
Rolling upgrades.....	711
The IBM Storage Scale FPO cluster.....	713
Failure detection.....	715
Disk Failures.....	716
Node failure.....	718
Handling multiple nodes failure.....	720
Network switch failure.....	721
Data locality.....	721
Disk Replacement.....	729
Auto recovery.....	731
Failure and recovery.....	732
QoS support for autorecovery.....	734
Restrictions.....	734
Chapter 49. Encryption.....	735
Encryption keys.....	735
Encryption policies.....	736
Encryption policy rules.....	736
Preparation for encryption.....	742
Establishing an encryption-enabled environment.....	747
Simplified setup: Using SKLM with a self-signed certificate.....	747
Setup using HashiCorp Vault KMIP Secrets Engine.....	757
Simplified setup: Using SKLM with a certificate chain.....	761
Simplified setup: Valid and invalid configurations.....	772

Simplified setup: Accessing a remote file system.....	774
Accessing an encrypted remote file system using keys from Vault KMIP Secrets Engine.....	778
Simplified setup: Doing other tasks.....	780
Regular setup: Using SKLM with a self-signed certificate.....	788
Regular setup: Using SKLM with a certificate chain.....	797
Regular setup: Accessing a remote file system.....	807
Converting encryption configuration from regular setup to simplified setup.....	807
Configuring encryption with SKLM 2.7 or later.....	810
Configuring encryption with the Thales Vormetric DSM key server.....	812
Configuring encryption with the Thales CipherTrust Manager key server by using a local certificate authority.....	821
Configuring encryption with the Thales CipherTrust Manager key server by using an external certificate authority.....	826
Certificate expiration warnings.....	832
Renewing client and server certificates.....	835
Certificate expiration dates and error messages.....	835
Renewing expired server certificates.....	838
Renewing expired client certificates.....	845
Encryption hints.....	856
Secure deletion.....	856
Key rotation: Replacing master encryption keys.....	858
Encryption and standards compliance.....	860
Encryption and FIPS 140-2 certification.....	860
Encryption and NIST SP800-131A compliance.....	861
Encryption in a multicluster environment.....	861
Encryption in a Disaster Recovery environment.....	862
Encryption and backup/restore.....	862
Encryption and snapshots.....	862
Encryption and a local read-only cache (LROC) device.....	862
Encryption and external pools.....	863
Encryption requirements and limitations.....	863
Chapter 50. Managing certificates to secure communications between GUI web server and web browsers.....	865
Chapter 51. Securing protocol data.....	867
Planning for protocol data security.....	869
Configuring protocol data security.....	869
Enabling secured connection between the IBM Storage Scale system and authentication server	870
Securing data transfer.....	873
Securing NFS data transfer.....	873
Securing SMB data transfer.....	875
Secured object data transfer.....	875
Data security limitations.....	875
Chapter 52. Cloud services: Transparent cloud tiering and Cloud data sharing.....	877
Administering files for Transparent cloud tiering.....	877
Applying a policy on a Transparent cloud tiering node.....	877
Migrating files to the cloud storage tier.....	880
Pre-migrating files to the cloud storage tier.....	880
Recalling files from the cloud storage tier.....	882
Reconciling files between IBM Storage Scale file system and cloud storage tier.....	882
Cleaning up files transferred to the cloud storage tier.....	883
Deleting cloud objects.....	884
Managing reversioned files.....	885
Listing files migrated to the cloud storage tier.....	885
Restoring files.....	886

Restoring Cloud services configuration.....	887
Checking the Cloud services database integrity.....	888
Manual recovery of Transparent cloud tiering database.....	888
Scale out backup and restore (SOBAR) for Cloud services.....	889
Cloud data sharing.....	902
Listing files exported to the cloud.....	903
Importing cloud objects exported through an old version of Cloud data sharing.....	905
Administering Transparent cloud tiering and Cloud data sharing services.....	906
Stopping Cloud services software.....	906
Monitoring the health of Cloud services software.....	906
Checking the Cloud services version.....	908
Known limitations of Cloud services	909
Chapter 53. Managing file audit logging.....	911
Managing the list of monitored events.....	911
Manage and list currently enabled audits of all types.....	911
Chapter 54. RDMA tuning.....	913
Chapter 55. Configuring Mellanox Memory Translation Table (MTT) for GPFS RDMA VERBS Operation.....	915
Chapter 56. Administering cloudkit.....	917
Mounting GPFS remote mount on compute nodes.....	917
Unmounting GPFS remote mount on compute nodes.....	918
Editing or scaling out an IBM Storage Scale cloud cluster.....	918
Enabling IBM Storage Scale GUI access by using JumpHost.....	919
Enabling and disabling repository access.....	920
Enabling AFM caching.....	921
Chapter 57. Administering AFM.....	923
Creating an AFM relationship by using the NFS protocol.....	923
Setting up the home cluster.....	923
Setting up the cache cluster.....	925
Example of creating an AFM relationship by using the NFS protocol	925
Example of AFM support for Kerberos-enabled NFS protocol exports.....	928
Creating an AFM relationship by using GPFS protocol.....	928
Setting up the home cluster.....	928
Setting up the cache cluster.....	929
Example of creating an AFM relationship by using the GPFS protocol.....	929
Checking the synchronization status of an AFM fileset.....	930
Pre-populating metadata by using the out of band prefetch.....	931
Chapter 58. Administering AFM DR.....	933
Enabling integrated archive manager (IAM) modes on AFM-DR filesets.....	933
Creating an AFM-based DR relationship.....	933
Converting GPFS filesets to AFM DR.....	935
Converting AFM relationship to AFM DR.....	936
Chapter 59. Administering AFM to cloud object storage.....	939
Managing AFM to cloud object storage keys.....	939
Creating AFM to cloud object storage relation in different modes.....	940
Evicting files or objects data.....	947
Evicting files or objects metadata.....	949
Evicting data or objects by using the manual updates mode of the AFM to cloud object storage.....	950
Mapping a directory to a cloud object storage bucket.....	951

Uploading objects.....	952
Downloading objects.....	954
Downloading objects by using the outband method	956
Converting IBM Storage Scale independent fileset to manual update mode.....	957
Uploading and downloading files from MU mode fileset.....	959
AFM to COS upload and download statistics.....	962
Synchronization of AFM to cloud object storage data to the bucket by using prefix.....	964
Migration of a transparent cloud tiering-enabled IBM Storage Scale fileset or file system to an AFM to cloud object storage fileset in the manual update mode.....	965
Promoting a TCT-enabled fileset to an AFM to cloud object storage fileset in the manual update mode.....	966
Promoting a TCT-enabled file system to an AFM to cloud object storage fileset in the manual update mode.....	968
Converting an AFM to cloud object storage fileset supporting Azure Blob storage by using a MinIO gateway to native Azure Blob storage.....	969
Chapter 60. Highly available write cache (HAWC).....	971
Applications that can benefit from HAWC.....	971
Restrictions and tuning recommendations for HAWC.....	972
Using HAWC.....	972
Chapter 61. Local read-only cache.....	975
Chapter 62. Miscellaneous advanced administration topics.....	977
Changing IP addresses or host names of cluster nodes.....	977
First scenario: All the nodes in the cluster are affected.....	977
Second scenario: Some of the nodes in the cluster are affected.....	978
Updating IP addresses or host names in other configurations.....	979
Enabling a cluster for IPv6.....	980
Using multiple token servers.....	981
Exporting file system definitions between clusters.....	981
IBM Storage Scale port usage.....	982
IBM Storage Scale GUI port usage.....	984
Securing the IBM Storage Scale system using firewall.....	984
Firewall recommendations for the IBM Storage Scale installation.....	985
Firewall recommendations for internal communication among nodes.....	985
Firewall recommendations for protocol access.....	987
Firewall recommendations for IBM Storage Scale GUI.....	991
Firewall recommendations for IBM SKLM.....	992
Firewall recommendations for Thales Vormetric Data Security Manager (DSM).....	993
Firewall recommendations for Performance Monitoring tool.....	994
Firewall considerations for Active File Management (AFM).....	994
Firewall considerations for remote mounting of file systems.....	995
Firewall recommendations for using IBM Storage Protect with IBM Storage Scale.....	995
Firewall considerations for using IBM Spectrum Archive with IBM Storage Scale.....	995
Firewall recommendations for call home.....	996
Examples of how to open firewall ports.....	996
Supported web browser versions and web browser settings for GUI.....	998
Chapter 63. GUI limitations.....	999
Accessibility features for IBM Storage Scale.....	1001
Accessibility features.....	1001
Keyboard navigation.....	1001
IBM and accessibility.....	1001

Notices.....	1003
Trademarks.....	1004
Terms and conditions for product documentation.....	1004
Glossary.....	1007
Index.....	1015

Tables

1. IBM Storage Scale library information units.....	xxii
2. Conventions.....	xli
3. Configuration attributes on the mmchconfig command.....	9
4. Recommendations for selecting value for the maxTcpConnsPerNodeConn parameter.....	78
5. Attributes and default values.....	103
6. Supported Components.....	104
7. Device details for Device 1.....	124
8. Default Configurations for INT_LOG_MAX_PAYLOAD_SIZE	124
9. Recommended Configurations for INT_LOG_MAX_PAYLOAD_SIZE Values.....	124
10. AFM, AFM-DR, and AFM to cloud object storage configuration parameters.....	133
11. NFS server parameters.....	170
12. Triple-O Heat Template Environment Parameters.....	185
13. IP pair table on node A.....	190
14. IP pair table on NSD server for an N:N connection model.....	191
15. IP pair table on NSD server for an M*N connection model.....	191
16. IP pair table on NSD server for multiple IP addresses.....	192
17. Compression libraries and their required file system format level and format number.....	231
18. COMPRESSION and illCompressed flags.....	235
19. Set QoS classes to unlimited.....	238
20. Allocate the available IOPS.....	239
21. File system format changes between versions of IBM Storage Scale 5.2.x.x and 5.1.x.x.....	267
22. Authentication requirements for each file access protocol.	328
23. Object services and protocol nodes.....	369

24. Object input behavior in unified_mode.....	387
25. Configuration options for [swift-constraints] in swift.conf.....	410
26. Configurable options for [DEFAULT] in object-server-sof.conf.....	412
27. Configurable options for [capabilities] in spectrum-scale-object.conf.....	414
28. Configuration options for [DEFAULT] in spectrum-scale-objectizer.conf.....	414
29. Configuration options for [IBMOBJECTIZER-LOGGER] in spectrum-scale-objectizer.conf.....	414
30. Configuration options for object-server.conf.....	415
31. Configuration options for /etc/sysconfig/memcached.....	415
32. Configuration options for proxy-server.conf.....	415
33. mkldap command parameters.....	457
34. Removal of a file with ACL entries DELETE and DELETE_CHILD.....	469
35. Mapping from NFSv4 ACL entry to SMB Security Descriptor.....	476
36. Mapping from SMB Security Descriptor to NFSv4 ACL entry with unixmap or ldapmap id mapping...477	477
37. Mapping from SMB Security Descriptor to NFSv4 ACL entry with default id mapping.....	478
38. ACL permissions required to work on files and directories, while using SMB protocol (table 1 of 2)..482	482
39. ACL permissions required to work on files and directories, while using SMB protocol (table 2 of 2)..483	483
40. ACL permissions required to work on files and directories, while using NFS protocol (table 1 of 2).. 484	484
41. ACL permissions required to work on files and directories, while using NFS protocol (table 2 of 2).. 484	484
42. Commands and reference to manage ACL tasks.....	487
43. ACL options that are available to manipulate object read ACLs.....	494
44. Summary of commands to set up cross-cluster file system access.....	515
45. Effects of options on uncompressed or compressed files.....	536
46. The effects of file operations on an immutable file or an appendOnly file.....	595
47. IAM modes and their effects on file operations on immutable files.....	596
48. Example for retention period.....	609

49. Example - Time stamp of snapshots that are retained based on the retention policy.....	609
50. User identification attributes.....	668
51. Group identification attribute.....	668
52. Valid EncParamString values.....	737
53. Valid combine parameter string values.....	738
54. Valid wrapping parameter string values.....	738
55. Required version of IBM Storage Scale.....	743
56. Remote Key Management servers.....	743
57. The RKM.conf file.....	745
58. The client keystore directory.....	746
59. Configuring the cluster for encryption in the simplified setup.....	751
60. Configuring the cluster to use HashiCorp Vault KMIP Secrets Engine.....	758
61. Configuring the cluster for encryption in the simplified setup.....	765
62. Setup of Cluster1 and Cluster2	774
63. Setup of Cluster1 and Cluster2	778
64. Managing another key server.....	783
65. Frequency of warnings.....	834
66. Comparing default lifetimes of key server and key client certificates.....	835
67. Security features that are used to secure authentication server.....	867
68. Sample policy list.....	879
69. Parameter description.....	899
70. Parameter description.....	900
71. Parameter description.....	901
72. Parameter description.....	901
73. Parameter description.....	901

74. Instance maximum limit.....	918
75. IBM Storage Scale port usage.....	982
76. Firewall related information.....	984
77. Recommended port numbers that can be used for installation.....	985
78. Recommended port numbers that can be used for internal communication.....	986
79. Recommended port numbers for NFS access.....	987
80. Recommended port numbers for SMB access.....	988
81. Recommended port numbers for the S3 access	988
82. Port numbers for object access.....	989
83. Port numbers for object authentication.....	990
84. Port numbers for Postgres database for object protocol.....	990
85. Consolidated list of recommended ports for different functions.....	990
86. Firewall recommendations for GUI.....	992
87. Firewall recommendations for GKLM.....	993
88. Firewall recommendations for DSM.....	993
89. Recommended port numbers that can be used for Performance Monitoring tool.....	994
90. Required port number for mmbackup and IBM Storage Protect for Space Management connectivity to IBM Spectrum Protect server.....	995
91. Recommended port numbers that can be used for call home.....	996

About this information

This edition applies to IBM Storage Scale version 5.2.1 for AIX®, Linux®, and Windows.

IBM Storage Scale is a file management infrastructure, based on IBM General Parallel File System (GPFS) technology, which provides unmatched performance and reliability with scalable access to critical file data.

To find out which version of IBM Storage Scale is running on a particular AIX node, enter:

```
lslpp -l gpfs\*
```

To find out which version of IBM Storage Scale is running on a particular Linux node, enter:

```
rpm -qa | grep gpfs      (for SLES and Red Hat Enterprise Linux)
```

```
dpkg -l | grep gpfs      (for Ubuntu Linux)
```

To find out which version of IBM Storage Scale is running on a particular Windows node, open **Programs and Features** in the control panel. The IBM Storage Scale installed program name includes the version number.

Which IBM Storage Scale information unit provides the information you need?

The IBM Storage Scale library consists of the information units listed in [Table 1 on page xxii](#).

To use these information units effectively, you must be familiar with IBM Storage Scale and the AIX, Linux, or Windows operating system, or all of them, depending on which operating systems are in use at your installation. Where necessary, these information units provide some background information relating to AIX, Linux, or Windows. However, more commonly they refer to the appropriate operating system documentation.

Note: Throughout this documentation, the term "Linux" refers to all supported distributions of Linux, unless otherwise specified.

Table 1. IBM Storage Scale library information units

Information unit	Type of information	Intended users
<i>IBM Storage Scale: Concepts, Planning, and Installation Guide</i>	<p>This guide provides the following information:</p> <p>Product overview</p> <ul style="list-style-type: none"> • Overview of IBM Storage Scale • GPFS architecture • Protocols support overview: Integration of protocol access methods with GPFS • Active File Management • AFM-based Asynchronous Disaster Recovery (AFM DR) • Introduction to AFM to cloud object storage • Introduction to system health and troubleshooting • Introduction to performance monitoring • Data protection and disaster recovery in IBM Storage Scale • Introduction to IBM Storage Scale GUI • IBM Storage Scale management API • Introduction to Cloud services • Introduction to file audit logging • Introduction to clustered watch folder • Understanding call home • IBM Storage Scale in an OpenStack cloud deployment • IBM Storage Scale product editions • IBM Storage Scale license designation • Capacity-based licensing • Dynamic pagepool 	System administrators, analysts, installers, planners, and programmers of IBM Storage Scale clusters who are very experienced with the operating systems on which each IBM Storage Scale cluster is based

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
<i>IBM Storage Scale: Concepts, Planning, and Installation Guide</i>	Planning <ul style="list-style-type: none">• Planning for GPFS• Planning for protocols• Planning for cloud services• Planning for IBM Storage Scale on Public Clouds• Planning for AFM• Planning for AFM DR• Planning for AFM to cloud object storage• Planning for performance monitoring tool• Planning for UEFI secure boot on x86_64 and secure boot on Linux on Z	
<i>IBM Storage Scale: Concepts, Planning, and Installation Guide</i>	<ul style="list-style-type: none">• Firewall recommendations• Considerations for GPFS applications• Security-Enhanced Linux support• Space requirements for call home data upload	

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
<i>IBM Storage Scale: Concepts, Planning, and Installation Guide</i>	<p>Installing</p> <ul style="list-style-type: none"> • Steps for establishing and starting your IBM Storage Scale cluster • Installing IBM Storage Scale on Linux nodes and deploying protocols • Installing IBM Storage Scale on public cloud by using cloudkit • Installing IBM Storage Scale on AIX nodes • Installing IBM Storage Scale on Windows nodes • Installing Cloud services on IBM Storage Scale nodes • Installing and configuring IBM Storage Scale management API • Installing GPUDirect Storage for IBM Storage Scale • Installation of Active File Management (AFM) • Installing AFM Disaster Recovery • Installing call home • Installing file audit logging • Installing clustered watch folder • Installing the signed kernel modules for UEFI secure boot on x86_64 • Installing the signed kernel modules for secure boot on Linux on Z • Steps to permanently uninstall IBM Storage Scale <p>Upgrading</p> <ul style="list-style-type: none"> • IBM Storage Scale supported upgrade paths • Online upgrade support for protocols and performance monitoring • Upgrading IBM Storage Scale nodes 	System administrators, analysts, installers, planners, and programmers of IBM Storage Scale clusters who are very experienced with the operating systems on which each IBM Storage Scale cluster is based

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
<i>IBM Storage Scale: Concepts, Planning, and Installation Guide</i>	<ul style="list-style-type: none"> • Upgrading IBM Storage Scale non-protocol Linux nodes • Upgrading IBM Storage Scale protocol nodes • Upgrading IBM Storage Scale on cloud • Upgrading GPUDirect Storage • Upgrading AFM and AFM DR • Upgrading object packages • Upgrading SMB packages • Upgrading NFS packages • Upgrading call home • Upgrading the performance monitoring tool • Upgrading signed kernel modules for UEFI secure boot on x86_64 and Linux on Z • Manually upgrading pmswift • Manually upgrading the IBM Storage Scale management GUI • Upgrading Cloud services • Upgrading to IBM Cloud Object Storage software level 3.7.2 and above • Upgrade paths and commands for file audit logging and clustered watch folder • Upgrading IBM Storage Scale components with the installation toolkit • Protocol authentication configuration changes during upgrade • Changing the IBM Storage Scale product edition • Completing the upgrade to a new level of IBM Storage Scale • Reverting to the previous level of IBM Storage Scale 	System administrators, analysts, installers, planners, and programmers of IBM Storage Scale clusters who are very experienced with the operating systems on which each IBM Storage Scale cluster is based

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
<i>IBM Storage Scale: Concepts, Planning, and Installation Guide</i>	<ul style="list-style-type: none">• Coexistence considerations• Compatibility considerations• Considerations for IBM Storage Protect for Space Management• Applying maintenance to your IBM Storage Scale system• Guidance for upgrading the operating system on IBM Storage Scale nodes• Considerations for upgrading from an operating system not supported in IBM Storage Scale 5.1.x.x• Servicing IBM Storage Scale protocol nodes• Offline upgrade with complete cluster shutdown	

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
<i>IBM Storage Scale: Administration Guide</i>	<p>This guide provides the following information:</p> <p>Configuring</p> <ul style="list-style-type: none"> • Configuring the GPFS cluster • Configuring GPUDirect Storage for IBM Storage Scale • Configuring the CES and protocol configuration • Configuring and tuning your system for GPFS • Parameters for performance tuning and optimization • Ensuring high availability of the GUI service • Configuring and tuning your system for Cloud services • Configuring IBM Power Systems for IBM Storage Scale • Configuring file audit logging • Configuring clustered watch folder • Configuring the cloudkit • Configuring Active File Management • Configuring AFM-based DR • Configuring AFM to cloud object storage • Tuning for Kernel NFS backend on AFM and AFM DR • Configuring call home • Integrating IBM Storage Scale Cinder driver with Red Hat OpenStack Platform 16.1 • Configuring Multi-Rail over TCP (MROT) • Dynamic pagepool configuration 	System administrators or programmers of IBM Storage Scale systems

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
<i>IBM Storage Scale: Administration Guide</i>	Administering <ul style="list-style-type: none">• Performing GPFS administration tasks• Performing parallel copy with mmxcp command• Protecting file data: IBM Storage Scale safeguarded copy• Verifying network operation with the mmnetverify command• Managing file systems• File system format changes between versions of IBM Storage Scale• Managing disks	System administrators or programmers of IBM Storage Scale systems

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
<i>IBM Storage Scale: Administration Guide</i>	<ul style="list-style-type: none"> • Managing protocol services • Managing protocol user authentication • Managing protocol data exports • Managing S3 protocol • Managing object storage • Managing GPFS quotas • Managing GUI users • Managing GPFS access control lists • Native NFS and GPFS • Accessing a remote GPFS file system • Information lifecycle management for IBM Storage Scale • Creating and maintaining snapshots of file systems • Creating and managing file clones • Scale Out Backup and Restore (SOBAR) • Data Mirroring and Replication • Implementing a clustered NFS environment on Linux • Implementing Cluster Export Services • Identity management on Windows / RFC 2307 Attributes • Protocols cluster disaster recovery • File Placement Optimizer • Encryption • Managing certificates to secure communications between GUI web server and web browsers • Securing protocol data • Managing file audit logging • RDMA tuning • Configuring Mellanox Memory Translation Table (MTT) for GPFS RDMA VERBS Operation • Administering cloudkit • Administering AFM • Administering AFM DR 	System administrators or programmers of IBM Storage Scale systems

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
<i>IBM Storage Scale: Administration Guide</i>	<ul style="list-style-type: none">• Administering AFM to cloud object storage• Highly available write cache (HAWC)• Local read-only cache• Miscellaneous advanced administration topics• GUI limitations	System administrators or programmers of IBM Storage Scale systems

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
<i>IBM Storage Scale: Problem Determination Guide</i>	<p>This guide provides the following information:</p> <p>Monitoring</p> <ul style="list-style-type: none"> • Monitoring system health by using IBM Storage Scale GUI • Monitoring system health by using the mmhealth command • Dynamic pagepool monitoring • Performance monitoring • Monitoring GPUDirect storage • Monitoring events through callbacks • Monitoring capacity through GUI • Monitoring AFM and AFM DR • Monitoring AFM to cloud object storage • GPFS SNMP support • Monitoring the IBM Storage Scale system by using call home • Monitoring remote cluster through GUI • Monitoring file audit logging • Monitoring clustered watch folder • Monitoring local read-only cache <p>Troubleshooting</p> <ul style="list-style-type: none"> • Best practices for troubleshooting • Understanding the system limitations • Collecting details of the issues • Managing deadlocks • Installation and configuration issues • Upgrade issues • CCR issues • Network issues • File system issues • Disk issues • GPUDirect Storage troubleshooting • Security issues • Protocol issues • Disaster recovery issues • Performance issues 	<p>System administrators of GPFS systems who are experienced with the subsystems used to manage disks and who are familiar with the concepts presented in the <i>IBM Storage Scale: Concepts, Planning, and Installation Guide</i></p>

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
<i>IBM Storage Scale: Problem Determination Guide</i>	<ul style="list-style-type: none">• GUI and monitoring issues• AFM issues• AFM DR issues• AFM to cloud object storage issues• Transparent cloud tiering issues• File audit logging issues• Cloudkit issues• Troubleshooting mmwatch• Maintenance procedures• Recovery procedures• Support for troubleshooting• References	

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
<i>IBM Storage Scale: Command and Programming Reference Guide</i>	<p>This guide provides the following information:</p> <p>Command reference</p> <ul style="list-style-type: none"> • cloudkit command • gpfs.snap command • mmaddcallback command • mmadddisk command • mmaddnode command • mmadquery command • mmafmconfig command • mmafmcosaccess command • mmafmcosconfig command • mmafmcosctl command • mmafmcoskeys command • mmafmctl command • mmafmlocal command • mmapplypolicy command • mmaudit command • mmauth command • mmbackup command • mmbackupconfig command • mmbuildgpl command • mmcachectl command • mmcallhome command • mmces command • mmchattr command • mmchcluster command • mmchconfig command • mmchdisk command • mmcheckquota command • mmchfileset command • mmchfs command • mmchlicense command • mmchmgr command • mmchnode command • mmchnodeclass command • mmchnsd command • mmchpolicy command • mmchpool command • mmchqos command • mmclidecode command 	<ul style="list-style-type: none"> • System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
<i>IBM Storage Scale: Command and Programming Reference Guide</i>	<ul style="list-style-type: none"> • mmclone command • mmccloudgateway command • mmcrcluster command • mmcrfileset command • mmcrfs command • mmcrnodeclass command • mmcrnsd command • mmcrsnapshot command • mmdefedquota command • mmdefquotaoff command • mmdefquotaon command • mmdefragfs command • mmdelacl command • mmdelcallback command • mmdeldisk command • mmdelfileset command • mmdeflfs command • mmdeinode command • mmdeinodeclass command • mmdeinsn command • mmdelesnapshot command • mmdf command • mmdiag command • mmddsh command • mmeditacl command • mmedquota command • mmexportfs command • mmfsck command • mmfsckx command • mmfsctl command • mmgetacl command • mmgetstate command • mmhadoopctl command • mmhdfs command • mmhealth command • mmimgbackup command • mmimgrestore command • mmimportfs command • mmkeyserv command 	<ul style="list-style-type: none"> • System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
<i>IBM Storage Scale: Command and Programming Reference Guide</i>	<ul style="list-style-type: none"> • mmlinkfileset command • mmIsattr command • mmIscallback command • mmIscluster command • mmIsconfig command • mmIsdisk command • mmIsfileset command • mmIsfs command • mmIslicense command • mmIsmgr command • mmIsmount command • mmIsnodeclass command • mmIsnsd command • mmIspolicy command • mmIspool command • mmIsqos command • mmIsquota command • mmIssnapshot command • mmMigratefs command • mmMount command • mmNetverify command • mmNfs command • mmNsdiscover command • mmObj command • mmPerfmon command • mmPmon command • mmProtocoltrace command • mmPsnap command • mmPstat command • mmPutacl command • mmQos command • mmQuotaoff command • mmQuotaon command • mmReclaimSpace command • mmRemotecluster command • mmRemoteefs command • mmRepquota command • mmRestoreconfig command • mmRestoreefs command • mmRestrictedctl command • mmRestripefile command 	<ul style="list-style-type: none"> • System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
<i>IBM Storage Scale: Command and Programming Reference Guide</i>	<ul style="list-style-type: none"> • mmrestripefs command • mmrpldisk command • mmsdrrestore command • mmsetquota command • mmshutdown command • mmsmb command • mmsnapdir command • mmstartup command • mmstartpolicy command • mms3 command • mmtracectl command • mmumount command • mmunlinkfileset command • mmuserauth command • mmwatch command • mmwinservctl command • mmxcp command • spectrumscale command <p>Programming reference</p> <ul style="list-style-type: none"> • IBM Storage Scale Data Management API for GPFS information • GPFS programming interfaces • GPFS user exits • IBM Storage Scale management API endpoints • Considerations for GPFS applications 	<ul style="list-style-type: none"> • System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
<i>IBM Storage Scale: Big Data and Analytics Guide</i>	<p>This guide provides the following information:</p> <ul style="list-style-type: none"> Summary of changes Big data and analytics support Hadoop Scale Storage Architecture <ul style="list-style-type: none"> • Elastic Storage Server • Erasure Code Edition • Share Storage (SAN-based storage) • File Placement Optimizer (FPO) • Deployment model • Additional supported storage features IBM Spectrum® Scale support for Hadoop <ul style="list-style-type: none"> • HDFS transparency overview • Supported IBM Storage Scale storage modes • Hadoop cluster planning • CES HDFS • Non-CES HDFS • Security • Advanced features • Hadoop distribution support • Limitations and differences from native HDFS • Problem determination IBM Storage Scale Hadoop performance tuning guide <ul style="list-style-type: none"> • Overview • Performance overview • Hadoop Performance Planning over IBM Storage Scale • Performance guide 	<ul style="list-style-type: none"> • System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
<i>IBM Storage Scale: Big Data and Analytics Guide</i>	Cloudera Data Platform (CDP) Private Cloud Base <ul style="list-style-type: none"> • Overview • Planning • Installing • Configuring • Administering • Monitoring • Upgrading • Limitations • Problem determination 	<ul style="list-style-type: none"> • System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard
<i>IBM Storage Scale: Big Data and Analytics Guide</i>	Cloudera HDP 3.X <ul style="list-style-type: none"> • Planning • Installation • Upgrading and uninstallation • Configuration • Administration • Limitations • Problem determination Open Source Apache Hadoop <ul style="list-style-type: none"> • Open Source Apache Hadoop without CES HDFS • Open Source Apache Hadoop with CES HDFS 	<ul style="list-style-type: none"> • System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
<i>IBM Storage Scale Erasure Code Edition Guide</i>	<p>IBM Storage Scale Erasure Code Edition</p> <ul style="list-style-type: none"> • Summary of changes • Introduction to IBM Storage Scale Erasure Code Edition • Planning for IBM Storage Scale Erasure Code Edition • Installing IBM Storage Scale Erasure Code Edition • Uninstalling IBM Storage Scale Erasure Code Edition • Creating an IBM Storage Scale Erasure Code Edition storage environment • Using IBM Storage Scale Erasure Code Edition for data mirroring and replication • Deploying IBM Storage Scale Erasure Code Edition on VMware infrastructure • Upgrading IBM Storage Scale Erasure Code Edition • Incorporating IBM Storage Scale Erasure Code Edition in an Elastic Storage Server (ESS) cluster • Incorporating IBM Elastic Storage Server (ESS) building block in an IBM Storage Scale Erasure Code Edition cluster • Administering IBM Storage Scale Erasure Code Edition • Troubleshooting • IBM Storage Scale RAID Administration 	<ul style="list-style-type: none"> • System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Table 1. IBM Storage Scale library information units (continued)

Information unit	Type of information	Intended users
IBM Storage Scale Container Native Storage Access	<p>This guide provides the following information:</p> <ul style="list-style-type: none"> • Overview • Planning • Installation prerequisites • Installing the IBM Storage Scale container native operator and cluster • Upgrading • Configuring IBM Storage Scale Container Storage Interface (CSI) driver • Using IBM Storage Scale GUI • Maintenance of a deployed cluster • Cleaning up the container native cluster • Monitoring • Troubleshooting • References 	<ul style="list-style-type: none"> • System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard
IBM Storage Scale Container Storage Interface Driver Guide	<p>This guide provides the following information:</p> <ul style="list-style-type: none"> • Summary of changes • Introduction • Planning • Installation • Upgrading • Configurations • Using IBM Storage Scale Container Storage Interface Driver • Managing IBM Storage Scale when used with IBM Storage Scale Container Storage Interface driver • Cleanup • Limitations • Troubleshooting 	<ul style="list-style-type: none"> • System administrators of IBM Storage Scale systems • Application programmers who are experienced with IBM Storage Scale systems and familiar with the terminology and concepts in the XDSM standard

Prerequisite and related information

For updates to this information, see [IBM Storage Scale in IBM Documentation](#).

For the latest support information, see the [IBM Storage Scale FAQ in IBM Documentation](#).

Conventions used in this information

Table 2 on page xli describes the typographic conventions used in this information. UNIX file name conventions are used throughout this information.

Note: Users of IBM Storage Scale for Windows must be aware that on Windows, UNIX-style file names need to be converted appropriately. For example, the GPFS cluster configuration data is stored in the /var/mmfs/gen/mmsdrfs file. On Windows, the UNIX namespace starts under the %SystemDrive%\cygwin64 directory, so the GPFS cluster configuration data is stored in the C:\cygwin64\var\mmfs\gen\mmsdrfs file.

Table 2. Conventions

Convention	Usage
bold	Bold words or characters represent system elements that you must use literally, such as commands, flags, values, and selected menu options. Depending on the context, bold typeface sometimes represents path names, directories, or file names.
bold <u>underlined</u>	<u>bold</u> <u>underlined</u> keywords are defaults. These take effect if you do not specify a different keyword.
constant width	Examples and information that the system displays appear in constant-width typeface. Depending on the context, constant-width typeface sometimes represents path names, directories, or file names.
<i>italic</i>	<i>Italic</i> words or characters represent variable values that you must supply. <i>Italics</i> are also used for information unit titles, for the first use of a glossary term, and for general emphasis in text.
<key>	Angle brackets (less-than and greater-than) enclose the name of a key on the keyboard. For example, <Enter> refers to the key on your terminal or workstation that is labeled with the word <i>Enter</i> .
\	In command examples, a backslash indicates that the command or coding example continues on the next line. For example:
	<pre>mkcondition -r IBM.FileSystem -e "PercentTotUsed > 90" \ -E "PercentTotUsed < 85" -m p "FileSystem space used"</pre>
{item}	Braces enclose a list from which you must choose an item in format and syntax descriptions.
[item]	Brackets enclose optional items in format and syntax descriptions.
<Ctrl-x>	The notation <Ctrl-x> indicates a control character sequence. For example, <Ctrl-c> means that you hold down the control key while pressing <c>.
<i>item...</i>	Ellipses indicate that you can repeat the preceding item one or more times.
	In <i>synopsis</i> statements, vertical lines separate a list of choices. In other words, a vertical line means <i>Or</i> . In the left margin of the document, vertical lines indicate technical changes to the information.

Note: CLI options that accept a list of option values delimit with a comma and no space between values. As an example, to display the state on three nodes use mmgetstate -N NodeA,NodeB,NodeC. Exceptions to this syntax are listed specifically within the command.

How to send your comments

Your feedback is important in helping us to produce accurate, high-quality information. If you have any comments about this information or any other IBM Storage Scale documentation, send your comments to the following e-mail address:

`mhvrcfs@us.ibm.com`

Include the publication title and order number, and, if applicable, the specific location of the information about which you have comments (for example, a page number or a table number).

To contact the IBM Storage Scale development organization, send your comments to the following e-mail address:

`scale@us.ibm.com`

Summary of changes

This topic summarizes changes to the IBM Storage Scale licensed program and the IBM Storage Scale library. Within each information unit in the library, a vertical line (|) to the left of text and illustrations indicates technical changes or additions that are made to the previous edition of the information.

Summary of changes for IBM Storage Scale 5.2.1 as updated, September 2024

This release of the IBM Storage Scale licensed program and the IBM Storage Scale library includes the following improvements. All improvements are available after an upgrade, unless otherwise specified.

- [Commands, data types, and programming APIs](#)
- [Messages](#)
- [Stabilized, deprecated, and discontinued features](#)

AFM and AFM DR-related changes

- Range eviction of data in cache allows administrators to provide offset and length pair to evict the data from AFM and AFM to cloud object storage filesets.
- Administrators now can enable NFSv4 at the AFM fileset level.
- The temporary directories that are needed for recovery or resynchronization of AFM to cloud object storage are now customizable. Meaning, administrators can use **afmRecoveryUseFset** (for fileset level) and **afmRecoveryDir** (for cluster level) to set specific temporary directories.

AFM to cloud object storage

- Administrators can configure automated eviction after upload or reconcile tasks.
- S3 can be used to access data managed by AFM. For more information, see *S3 support overview* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Big data and analytics changes

For information on changes in IBM Storage Scale Big Data and Analytics support and HDFS protocol, see [Big Data and Analytics - summary of changes](#).

IBM Storage Scale Erasure Code Edition changes

For information on changes in the IBM Storage Scale Erasure Code Edition, see [IBM Storage Scale Erasure Code Edition - Summary of changes](#).

Cloudkit changes

- Azure Cloud Services is added to the public clouds where cloudkit can be used.
- Active file management (AFM) to cloud object storage (COS) is now offered for Google Cloud Platform (GCP).
- As a technology preview, the Cloudkit CLI offers support for AFM-COS setup by using Azure Blob Storage and Azure IBM Storage Scale clusters, see the following IBM Storage Scale support page: <https://www.ibm.com/support/pages/node/7165571>.
- As a technology preview, the Cloudkit CLI offers support for CES and protocols configuration on AWS. For more information, see the following IBM Storage Scale support page: <https://www.ibm.com/support/pages/node/7165478>.

File system core improvements

- From version 5.2.0.1 onward, IBM Storage Scale can use IBM Key Protect with Key Management Interoperability Protocol (KMIP) as a back-end server for file system encryption. The IBM Key Project with KMIP is available as a service in IBM Cloud and, by using the *regular setup* in the *IBM*

Storage Scale: Administration Guide, can be configured as a back-end key server for IBM Storage Scale for file system encryption.

- Starting with the file system format level 35.00 (available with IBM Storage Scale 5.2.1.0), the *mmaudit command* in the *IBM Storage Scale: Command and Programming Reference Guide* can provide information about the CLOSEWRITE events. This is a new event type that gets logged when a file is opened for a writing operation and then closed.
- NVMe persistent reservations for multi-attach volumes in Amazon Web Services (AWS). Added support for NVMe reservations for improved failover times, similar to SCSI persistent reservations (PR). This support is limited to multi-attach volumes in AWS virtual machines.
- From version 5.2.1 onward, nodes that are expelled by using the **mmexpelnod**e command are not allowed to rejoin the cluster until they are removed from the list by using **mmexpelnod**e **-r/---reset** (unless the **-o/-once** option is specified). In previous levels of IBM Storage Scale, expelled nodes were allowed to rejoin if the node appointed as cluster manager changed. Now, nodes that are expelled with **mmexpelnod**e stay expelled, even if the cluster manager goes down or otherwise changes. The new configuration can be reverted by issuing the **mmchconfig disablePersistExpelList=yes** command. For more information, see the *mmchconfig command* and the *mmexpelnod*e command in the *IBM Storage Scale: Command and Programming Reference Guide*.
- Improved administration options for expiring or expired cluster keys. Administrators can ease their planning of new keys generation and commit by harnessing a new field and a new option that are introduced in version 5.2.1.0. Supported by **mmauth show** and **mmremotecluster show** commands, the Key_Expiration field displays the expiration date for a key. And, if the cluster keys have expired by the time they are consulted, the **mmauth** command can be used with the **--force** option to generate new cluster keys. For more information, see the *mmauth command* and the *mmremotecluster command* in the *IBM Storage Scale: Command and Programming Reference Guide*.
- Improved performance in file creation or deletion during an **mmfsckx** command scanning of large reserved files. For more information, see *mmfsckx command* in the *IBM Storage Scale: Command and Programming Reference Guide*.
- The NSD server functionality is now supported on arm64 platforms.
- Optimized the compatibility mode for GPU direct storage to contribute to improve performance. For more information, see the *GPUDirect Storage support for IBM Storage Scale* section in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
- Improved logic to better handle IPv6/IPv4 mixed multi-cluster environment, which permits an IPv6 remote cluster and an IPv4 remote cluster both join an IPv6-enabled home cluster in some conditions.

File system protocol changes

- IBM Storage Scale 5.2.1. supports the S3 protocol.
- Updated Samba on IBM Storage Scale to the upstream Samba 4.19 version.
- By default, the **server signing** SMB configuration parameter is set to **mandatory**. For the SMB2 protocol, by design, signing cannot be disabled. If this parameter value is already set before the upgrade, the previously set value does not change after the upgrade. However, if you want to set other value, you can set the value after the upgrade, and services need to be restarted.

Important: An old install that did not set **server signing** will change and have **mandatory** turned on after update.

For more information, see the *mmsmb command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

- Minimum GnuTLS 3.6.13 is required for SMB. GnuTLS 3.7.2 is required on systems without the Linux `getrandom()`. These GnuTLS versions are present on the minimum supported OS releases.
- The time that is taken by the **mmsmb exportacl** command for a large number of SMB exports is improved significantly.

Installation toolkit changes

- Supports installation of S3 protocol packages on CES nodes.

Management API changes

The following endpoints are modified:

For more information, see the topic *IBM Storage Scale management API endpoints* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Linux on Z changes

- Starting with IBM Storage Scale 5.2.1, signed kernel modules for secure boot can be used on Linux on Z within environments that use RHEL 9.4. The `gpfs.bin.rpm` file that contains the signed kernel modules and the public key must be downloaded from Fix Central. For more information, see *Signed kernel modules for UEFI secure boot on x86_64 and secure boot Linux on Z* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
- Added SMC-D with SELinux enforcing mode. For more information, see detailed substeps in the *Configuring shared memory communications direct* section in the *IBM Storage Scale: Administration Guide*.
- Added SMC-D connections monitoring functionality into the **mmhealth** NETWORK component. For more information, see the *Monitoring the health of a node* section in the *IBM Storage Scale: Problem Determination Guide*.
- IBM Storage Scale 5.2.1 supports S3 on Linux on Z.

Note: Multiprotocol data access with S3 is not yet supported with Linux on Z.

S3 protocol

IBM Storage Scale 5.2.1 supports the S3 (Simple Storage Service) access protocol. It provides high-performance and scalable S3 object access to data, which is stored in IBM Storage Scale file systems. It is integrated with Cluster Export Services (CES) to provide highly available S3 access service in an IBM Storage Scale cluster.

Note: There is no upgrade support for S3 protocol (technical preview) shipped with IBM Storage Scale 5.2.0.x to S3 protocol shipped with IBM Storage Scale 5.2.1.x and later.

Native REST API (technology preview)

The native REST API is available in IBM Storage Scale as a technology preview. The feature adds a new control plane component to the IBM Storage Scale stack for administering clusters. The native REST API is an alternative for administering IBM Storage Scale clusters through the mm-command layer. The native REST API also adds a few security enhancements. For more information, see the following IBM Storage Scale support page: <https://www.ibm.com/support/pages/node/7165471>.

Commands, data types, and programming APIs

The following section lists the modifications to the documented commands, structures, and subroutines:

New commands

- **mms3**
- **mmpstat**

New structures

There are no new structure changes.

New subroutines

There are no new subroutines.

New user exits

There are no new user exits.

Changed commands

- **cloudkit**
- **mmaddcallback**

- **mmafmcosctl**
- **mmafmctl**
- **mmauth**
- **mmces**
- **mmchconfig**
- **mmcrlfileset**
- **mmchfileset**
- **mmexpelnode**
- **mmhealth**
- **mmimportnvmeof**
- **mmkeyserver**
- **mmperfmon**
- **mmremotecluster**
- **mmsmb**
- **spectrumscale**

Changed structures

There are no changed structures.

Changed subroutines

There are no changed subroutines.

Deleted commands

There are no deleted commands.

Deleted structures

There are no deleted structures.

Deleted subroutines

There are no deleted subroutines.

Messages

The following are the new, changed, and deleted messages:

New messages

6027-2577, 6027-3957, and 6027-3958

Changed messages

There are no changed messages.

Deleted messages

There are no deleted messages.

Chapter 1. Configuring the GPFS cluster

There are several tasks that are involved in managing your GPFS cluster. This topic points you to the information you need to get started.

GPFS cluster management tasks include the following topics:

- [“Creating your GPFS cluster” on page 1](#)
- [“Displaying cluster configuration information” on page 2](#)
- [“Specifying nodes as input to GPFS commands” on page 201](#)
- [“Adding nodes to a GPFS cluster” on page 4](#)
- [“Deleting nodes from a GPFS cluster” on page 5](#)
- [“Changing the GPFS cluster configuration data” on page 8](#)
- [“Cluster quorum with quorum nodes” on page 30](#)
- [“Cluster quorum with quorum nodes and tiebreaker disks” on page 31](#)
- [“Displaying and changing the file system manager node” on page 33](#)
- [“Determining how long mmrestripefs takes to complete” on page 204](#)
- [“Starting and stopping GPFS” on page 34](#)

Note: In IBM Storage Scale, many of these tasks can also be handled by the installation toolkit configuration options. For more information, see *Using the spectrumscale installation toolkit to perform installation tasks: Explanations and examples* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

For information on RAID administration, see *IBM Storage Scale RAID: Administration*.

Creating your GPFS cluster

You must first create a GPFS cluster by issuing the **mmcrcluster** command.

For more information, see the **mmcrcluster command** in *IBM Storage Scale: Command and Programming Reference Guide*.

For detailed information about how GPFS clusters are created and used, see *GPFS cluster creation considerations* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Related concepts

[Displaying cluster configuration information](#)

Use the **mmlscluster** command to display cluster configuration information.

[Security mode](#)

The security mode of a cluster determines the level of security that the cluster provides for communications between nodes in the cluster and also for communications between clusters.

[Minimum release level of a cluster](#)

The minimum release level of a cluster is the currently enabled level of functionality of the cluster. It is expressed as an IBM Storage Scale version number, such as 5.1.7.

[Cluster quorum with quorum nodes](#)

Cluster quorum defines the minimum number of GPFS quorum nodes that must have the GPFS daemon actively running on them for the cluster to be operational. This number is half of the defined GPFS quorum nodes plus one, so it is a good idea to have an odd number of GPFS quorum nodes defined. Cluster quorum using only GPFS quorum nodes is the default quorum algorithm.

[Cluster quorum with quorum nodes and tiebreaker disks](#)

To use cluster quorum with quorum nodes and tiebreaker disks, more requirements apply.

[Configuring cluster configuration repository](#)

The cluster configuration repository (CCR) is an internal configuration store for various IBM Storage Scale components. It is not meant to be used directly by the customer.

Related tasks

[Adding nodes to a GPFS cluster](#)

You can add nodes to an existing GPFS cluster by issuing the **mmaddnode** command or by using the IBM Storage Scale installation toolkit. The new nodes are available immediately after the successful completion of the process.

[Deleting nodes from a GPFS cluster](#)

You can delete nodes from a GPFS cluster by issuing the **mmdeinode** command.

[Changing the GPFS cluster configuration data](#)

You can use the **mmchcluster** or **mmchconfig** command to change the configuration attributes.

[Running IBM Storage Scale commands without remote root login](#)

With sudo wrapper scripts you can avoid configuring nodes to allow remote root login.

[Displaying and changing the file system manager node](#)

In general, GPFS performs the same functions on all nodes. There are also cases where one node provides a more global function that affects the operation of multiple nodes. For example, each file system is assigned a node that functions as a file system manager.

[Starting and stopping GPFS](#)

You can use the **mmstartup** and **mmshutdown** commands to start and stop GPFS on new or existing clusters.

[Shutting down an IBM Storage Scale cluster](#)

You might need to shut down an IBM Storage Scale cluster in an emergency.

Displaying cluster configuration information

Use the **mmlscluster** command to display cluster configuration information.

[“Basic configuration information” on page 2](#)

[“Information about protocol nodes” on page 3](#)

For more information, see **mmlscluster command** in *IBM Storage Scale: Command and Programming Reference Guide*.

Basic configuration information

To display basic cluster configuration information, enter the following command with no parameters:

```
mmlscluster
```

The command displays information like the following example:

```
GPFS cluster information
=====
  GPFS cluster name:      cluster1.kgn.ibm.com
  GPFS cluster id:       680681562214606028
  GPFS UID domain:      cluster1.kgn.ibm.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   k164n04.kgn.ibm.com    198.117.68.68   k164n04.kgn.ibm.com  quorum
    2   k164n05.kgn.ibm.com    198.117.68.69   k164n05.kgn.ibm.com  quorum
    3   k164n06.kgn.ibm.com    198.117.68.70   k164n06.kgn.ibm.com  quorum-manager
```

If the cluster uses a server-based repository, the command also displays the following information:

- The primary GPFS cluster configuration server
- The secondary GPFS cluster configuration server

Information about protocol nodes

To display information about the protocol nodes, enter the following command:

```
mmlscluster --ces
```

The command displays information like the following example:

```
GPFS cluster information
=====
GPFS cluster name: cluster1.kgn.ibm.com
GPFS cluster id: 4708497829760395040

Cluster Export Services global parameters
-----
Shared root directory: /gpfs/ces/ces
Enabled Services: OBJ SMB NFS
Log level: 0
Address distribution policy: even-coverage

Node Daemon node name IP address CES IP address list
-----
4 K16n07.kgn.ibm.com 192.168.4.4 10.18.64.23
5 K16n08.kgn.ibm.com 192.168.4.5 10.18.64.24
6 K16n09.kgn.ibm.com 192.168.4.6 10.18.64.26
7 K16n10.kgn.ibm.com 192.168.4.11 Node suspended, Node starting up
8 K16n11.kgn.ibm.com 192.168.4.12 Node suspended, Node starting up
```

Related concepts

Security mode

The security mode of a cluster determines the level of security that the cluster provides for communications between nodes in the cluster and also for communications between clusters.

Minimum release level of a cluster

The minimum release level of a cluster is the currently enabled level of functionality of the cluster. It is expressed as an IBM Storage Scale version number, such as 5.1.7.

Cluster quorum with quorum nodes

Cluster quorum defines the minimum number of GPFS quorum nodes that must have the GPFS daemon actively running on them for the cluster to be operational. This number is half of the defined GPFS quorum nodes plus one, so it is a good idea to have an odd number of GPFS quorum nodes defined. Cluster quorum using only GPFS quorum nodes is the default quorum algorithm.

Cluster quorum with quorum nodes and tiebreaker disks

To use cluster quorum with quorum nodes and tiebreaker disks, more requirements apply.

Configuring cluster configuration repository

The cluster configuration repository (CCR) is an internal configuration store for various IBM Storage Scale components. It is not meant to be used directly by the customer.

Related tasks

Creating your GPFS cluster

You must first create a GPFS cluster by issuing the **mmcrcluster** command.

Adding nodes to a GPFS cluster

You can add nodes to an existing GPFS cluster by issuing the **mmaddnode** command or by using the IBM Storage Scale installation toolkit. The new nodes are available immediately after the successful completion of the process.

Deleting nodes from a GPFS cluster

You can delete nodes from a GPFS cluster by issuing the **mmde1node** command.

Changing the GPFS cluster configuration data

You can use the **mmchcluster** or **mmchconfig** command to change the configuration attributes.

Running IBM Storage Scale commands without remote root login

With sudo wrapper scripts you can avoid configuring nodes to allow remote root login.

Displaying and changing the file system manager node

In general, GPFS performs the same functions on all nodes. There are also cases where one node provides a more global function that affects the operation of multiple nodes. For example, each file system is assigned a node that functions as a file system manager.

Starting and stopping GPFS

You can use the **mmstartup** and **mmshutdown** commands to start and stop GPFS on new or existing clusters.

Shutting down an IBM Storage Scale cluster

You might need to shut down an IBM Storage Scale cluster in an emergency.

Adding nodes to a GPFS cluster

You can add nodes to an existing GPFS cluster by issuing the **mmaddnode** command or by using the IBM Storage Scale installation toolkit. The new nodes are available immediately after the successful completion of the process.

Note: This topic provides details about how to add a node in the cluster by using the **mmaddnode** command. For more information about how to add a node to the cluster by using the installation toolkit, see *Adding nodes, NSDs, or file systems to an installation process* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

You must follow these rules when adding nodes to a GPFS cluster:

- You can issue the command only from a node that already belongs to the GPFS cluster.
- A node can belong to only one GPFS cluster at a time.
- IBM Storage Scale must be installed on the nodes before you run the **mmaddnode** command to add those nodes to the cluster. If you add a node by using the installation toolkit, you do not need to manually install IBM Storage Scale before you run the installation toolkit.
- A node must be available for the command to be successful. If any of the nodes listed are not available when the command is issued, a message listing those nodes is displayed. You must correct the problem on each node and reissue the command to add those nodes.
- To designate an IBM Storage Scale license to a node, you can use either of two methods:
 - Designate a license to the node with the **mmaddnode** command at the same time that you add the node to the cluster. For more information, see the *mmaddnode command* in the *IBM Storage Scale: Command and Programming Reference Guide*.
 - Designate a license to the node with the **mmchlicense** command after you add it to the cluster. For more information, see the *mmchlicense command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

To add node k164n01.kgn.ibm.com to the GPFS cluster, issue the following command:

```
mmaddnode -N k164n01.kgn.ibm.com
```

The system displays information similar to the following:

```
Mon Aug  9 21:53:30 EDT 2004: 6027-1664 mmaddnode: Processing node k164n01.kgn.ibm.com
mmaddnode: Command successfully completed
mmaddnode: 6027-1371 Propagating the cluster configuration data to all
          affected nodes.  This is an asynchronous process.
```

To confirm the addition of the nodes, issue the **mm1scluster** command.

The command displays information similar to the following:

```
GPFS cluster information
=====
GPFS cluster name:      cluster1.kgn.ibm.com
GPFS cluster id:        15529849231188177215
GPFS UID domain:       cluster1.kgn.ibm.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	k164n01.kgn.ibm.com	198.117.68.66	k164n01.kgn.ibm.com	
2	k164n02.kgn.ibm.com	198.117.68.67	k164n02.kgn.ibm.com	
3	k164n03.kgn.ibm.com	198.117.68.68	k164n03.kgn.ibm.com	quorum
4	k164n04.kgn.ibm.com	198.117.68.69	k164n04.kgn.ibm.com	quorum
3	k164n05.kgn.ibm.com	198.117.68.70	k164n05.kgn.ibm.com	quorum-manager

For more information, see **[mmaddnode command](#)**, **[mmlscluster command](#)** and **[mmchlicense command](#)** in *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

[Displaying cluster configuration information](#)

Use the **mmlscluster** command to display cluster configuration information.

[Security mode](#)

The security mode of a cluster determines the level of security that the cluster provides for communications between nodes in the cluster and also for communications between clusters.

[Minimum release level of a cluster](#)

The minimum release level of a cluster is the currently enabled level of functionality of the cluster. It is expressed as an IBM Storage Scale version number, such as 5.1.7.

[Cluster quorum with quorum nodes](#)

Cluster quorum defines the minimum number of GPFS quorum nodes that must have the GPFS daemon actively running on them for the cluster to be operational. This number is half of the defined GPFS quorum nodes plus one, so it is a good idea to have an odd number of GPFS quorum nodes defined. Cluster quorum using only GPFS quorum nodes is the default quorum algorithm.

[Cluster quorum with quorum nodes and tiebreaker disks](#)

To use cluster quorum with quorum nodes and tiebreaker disks, more requirements apply.

[Configuring cluster configuration repository](#)

The cluster configuration repository (CCR) is an internal configuration store for various IBM Storage Scale components. It is not meant to be used directly by the customer.

Related tasks

[Creating your GPFS cluster](#)

You must first create a GPFS cluster by issuing the **mmcrccluster** command.

[Deleting nodes from a GPFS cluster](#)

You can delete nodes from a GPFS cluster by issuing the **mmdeinode** command.

[Changing the GPFS cluster configuration data](#)

You can use the **mmchcluster** or **mmchconfig** command to change the configuration attributes.

[Running IBM Storage Scale commands without remote root login](#)

With sudo wrapper scripts you can avoid configuring nodes to allow remote root login.

[Displaying and changing the file system manager node](#)

In general, GPFS performs the same functions on all nodes. There are also cases where one node provides a more global function that affects the operation of multiple nodes. For example, each file system is assigned a node that functions as a file system manager.

[Starting and stopping GPFS](#)

You can use the **mmstartup** and **mmshutdown** commands to start and stop GPFS on new or existing clusters.

[Shutting down an IBM Storage Scale cluster](#)

You might need to shut down an IBM Storage Scale cluster in an emergency.

Deleting nodes from a GPFS cluster

You can delete nodes from a GPFS cluster by issuing the **mmdeinode** command.

The GPFS daemon must be shut down on a node before the node can be deleted. The following types of nodes cannot be deleted unless some reconfiguration is done:

- A node that is an NSD server cannot be deleted if it is the only NSD server for one or more NSDs in the cluster. Issue the **mmlsnsd** command to list the NSD servers and NSDs in the cluster. If a node is the only NSD server for some NSDs, you can issue the **mmchnsd** command to assign other NSD servers to those NSDs.
- A node that is a primary or secondary cluster configuration server cannot be deleted. Issue the **mmlscluster** command to list the primary and secondary cluster configuration servers, if any are configured. If a node is a primary or secondary configuration server, you can issue the **mmchcluster** command to create a new primary or secondary configuration server.
- If the GPFS state is *unknown* and the node is reachable on the network.

You cannot delete a node if both of the following are true:

- The node responds to a TCP/IP ping command from another node.
- The status of the node shows *unknown* when you use the **mmgetstate** command from another node in the cluster.

Note: You can delete such a node if you physically power it off.

- If the node is configured as a performance monitoring collector. In such cases, you need to remove the node from the performance monitoring configuration by using the **mpperfmon config update --collectors** command before deleting the node. Deleting a collector node causes loss of all the collected perfmon stats data on the collector node.
- If the node is defined as a Transparent cloud tiering node. You can determine whether a node is a Transparent cloud tiering node by issuing the **mmcloudgateway node list** command. If the node is listed as the Transparent cloud tiering node, and you still want to delete it without deleting the cluster, first use the **mmchnode** command to disable the Transparent cloud tiering node role:
 - If the node is a Transparent cloud tiering node, disable Transparent cloud tiering from the node by using the **mmchnode --cloud-gateway-disable** command, and then uninstall the Transparent cloud tiering rpms. Doing so ensures that the **mmdelnode** command does not fail on a Transparent cloud tiering node.

1. To delete the nodes listed in a file called `nodes_to_delete`, issue the following command:

```
mmdelnode -N /tmp/nodes_to_delete
```

where `nodes_to_delete` contains the nodes `k164n01` and `k164n02`. The system displays information similar to the following:

```
Verifying GPFS is stopped on all affected nodes ...
mmdelnode: Command successfully completed
mmdelnode: 6027-1371 Propagating the cluster configuration data to all
          affected nodes. This is an asynchronous process.
```

2. To confirm the deletion of the nodes, issue the following command:

```
mmlscluster
```

The system displays information similar to following:

GPFS cluster information				
=====				
GPFS cluster name:	cluster1.kgn.ibm.com			
GPFS cluster id:	15529849231188177215			
GPFS UID domain:	cluster1.kgn.ibm.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	k164n03.kgn.ibm.com	198.117.68.68	k164n03.kgn.ibm.com	quorum
2	k164n04.kgn.ibm.com	198.117.68.69	k164n04.kgn.ibm.com	quorum
3	k164n05.kgn.ibm.com	198.117.68.70	k164n05.kgn.ibm.com	quorum-manager

3. If you disabled file audit logging in step 1, you can enable it by following the instructions in “[Enabling file audit logging on a file system](#)” on page 125.

For information about deleting protocol nodes (CES nodes) from a cluster, see “[Deleting a Cluster Export Services node from an IBM Storage Scale cluster](#)” on page 61.

For more information, see **`mmdeinode command`** and **`mmlscluster command`** in *IBM Storage Scale: Command and Programming Reference Guide*.

Exercise caution when shutting down GPFS on quorum nodes or deleting quorum nodes from the GPFS cluster. If the number of remaining quorum nodes falls below the requirement for a quorum, then you are unable to perform file system operations. For more information about quorum, see *Quorum*, in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Related concepts

[Displaying cluster configuration information](#)

Use the **`mmlscluster`** command to display cluster configuration information.

[Security mode](#)

The security mode of a cluster determines the level of security that the cluster provides for communications between nodes in the cluster and also for communications between clusters.

[Minimum release level of a cluster](#)

The minimum release level of a cluster is the currently enabled level of functionality of the cluster. It is expressed as an IBM Storage Scale version number, such as 5.1.7.

[Cluster quorum with quorum nodes](#)

Cluster quorum defines the minimum number of GPFS quorum nodes that must have the GPFS daemon actively running on them for the cluster to be operational. This number is half of the defined GPFS quorum nodes plus one, so it is a good idea to have an odd number of GPFS quorum nodes defined. Cluster quorum using only GPFS quorum nodes is the default quorum algorithm.

[Cluster quorum with quorum nodes and tiebreaker disks](#)

To use cluster quorum with quorum nodes and tiebreaker disks, more requirements apply.

[Configuring cluster configuration repository](#)

The cluster configuration repository (CCR) is an internal configuration store for various IBM Storage Scale components. It is not meant to be used directly by the customer.

Related tasks

[Creating your GPFS cluster](#)

You must first create a GPFS cluster by issuing the **`mmcrccluster`** command.

[Adding nodes to a GPFS cluster](#)

You can add nodes to an existing GPFS cluster by issuing the **`mmaddnode`** command or by using the IBM Storage Scale installation toolkit. The new nodes are available immediately after the successful completion of the process.

[Changing the GPFS cluster configuration data](#)

You can use the **`mmchcluster`** or **`mmchconfig`** command to change the configuration attributes.

[Running IBM Storage Scale commands without remote root login](#)

With sudo wrapper scripts you can avoid configuring nodes to allow remote root login.

[Displaying and changing the file system manager node](#)

In general, GPFS performs the same functions on all nodes. There are also cases where one node provides a more global function that affects the operation of multiple nodes. For example, each file system is assigned a node that functions as a file system manager.

[Starting and stopping GPFS](#)

You can use the **`mmstartup`** and **`mmshutdown`** commands to start and stop GPFS on new or existing clusters.

[Shutting down an IBM Storage Scale cluster](#)

You might need to shut down an IBM Storage Scale cluster in an emergency.

Changing the GPFS cluster configuration data

You can use the **mmchcluster** or **mmchconfig** command to change the configuration attributes.

After you have configured the GPFS cluster, you can change configuration attributes with the **mmchcluster** command or the **mmchconfig** command. For more information, see the following topics:

- *mmchcluster command in IBM Storage Scale: Command and Programming Reference Guide*
- *mmchconfig command in IBM Storage Scale: Command and Programming Reference Guide*

Use the **mmchcluster** command to do the following tasks:

- Change the name of the cluster.
- Change the remote shell and remote file copy programs to be used by the nodes in the cluster. These commands must adhere to the syntax forms of the **ssh** and **scp** commands, but may implement an alternate authentication mechanism.
- Enable or disable the cluster configuration repository (CCR). For more information, see the *Cluster configuration data files* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

If you are using the traditional server-based (non-CCR) configuration repository, you can also do the following tasks:

- Change the primary or secondary GPFS cluster configuration server nodes. The primary or secondary server may be changed to another node in the GPFS cluster. That node must be available for the command to be successful.



Attention: If during the change to a new primary or secondary GPFS cluster configuration server, one or both of the old server nodes are down, it is imperative that you run the **mmchcluster -p LATEST** command as soon as the old servers are brought back online. Failure to do so may lead to disruption in GPFS operations.

- Synchronize the primary GPFS cluster configuration server node. If an invocation of the **mmchcluster** command fails, then you are prompted to reissue the command and specify LATEST on the -p option to synchronize all of the nodes in the GPFS cluster. Synchronization instructs all nodes in the GPFS cluster to use the most recently specified primary GPFS cluster configuration server.

For example, to change the primary server for the GPFS cluster data, enter:

```
mmchcluster -p k164n06
```

The system displays information similar to:

```
mmchcluster -p k164n06
mmchcluster: Command successfully completed
```

To confirm the change, issue the **mmlscluster** command.

The system displays information similar to:

```
GPFS cluster information
=====
GPFS cluster name:      cluster1.kgn.ibm.com
GPFS cluster id:        680681562214606028
GPFS UID domain:       cluster1.kgn.ibm.com
Remote shell command:   /usr/bin/ssh
Remote file copy command: /usr/bin/scp
Repository type:        server-based

GPFS cluster configuration servers:
-----
Primary server:    k164sn06.kgn.ibm.com
Secondary server:  k164n05.kgn.ibm.com
```

Node	Daemon node name	IP address	Admin node name	Designation
1	k164n04.kgn.ibm.com	198.117.68.68	k164n04.kgn.ibm.com	quorum
2	k164n05.kgn.ibm.com	198.117.68.69	k164n05.kgn.ibm.com	quorum
3	k164n06.kgn.ibm.com	198.117.68.70	k164n06.kgn.ibm.com	quorum-manager

 **Attention:** The **mmchcluster** command, when issued with either the -p or -s option, is designed to operate in an environment where the current primary and secondary GPFS cluster configuration servers are not available. As a result, the command can run without obtaining its regular serialization locks. To assure smooth transition to a new cluster configuration server, no other GPFS commands must be running when the command is issued nor should any other command be issued until the **mmchcluster** command has successfully completed.

For more information, see **mmchcluster command** and **mmlscluster command** in *IBM Storage Scale: Command and Programming Reference Guide*.

You might be able to tune your cluster for better performance by reconfiguring one or more attribute. Before you change any attribute, consider how the changes affect the operation of GPFS. For more information, see [Chapter 5, “Parameters for performance tuning and optimization,” on page 75](#).

Table 3 on page 9 details the GPFS cluster configuration attributes which can be changed by issuing the **mmchconfig** command. Variations under which these changes take effect are noted:

1. Take effect immediately and are permanent (-i).
2. Take effect immediately but do not persist when GPFS is restarted (-I).
3. Require that the GPFS daemon be stopped on all nodes for the change to take effect.
4. May be applied to only a subset of the nodes in the cluster.

For more information on the release history of tuning parameters, see [“Tuning parameters change history” on page 79](#).

Table 3. Configuration attributes on the **mmchconfig** command

Attribute name and Description	-i option allowed	-I option allowed	GPFS must be stopped on all nodes	List of NodeNames allowed	Change takes effect
adminMode Controls password-less access	yes	no	no	no	immediately
atimeDeferredSeconds Update behavior of atime when relatime is enabled	yes	yes	no	yes	if not immediately, on restart of the daemon
autoload Starts GPFS automatically	no	no	no	yes	on reboot of each node
automountDir Name of the automount directory	no	no	yes	no	on restart of the daemon
cesSharedRoot A directory to be used by the CES subsystem.	no	no	yes (on all CES nodes)	no	immediately

Table 3. Configuration attributes on the **mmchconfig** command (continued)

Attribute name and Description	-i option allowed	-I option allowed	GPFS must be stopped on all nodes	List of NodeNames allowed	Change takes effect
cipherList The security mode of the cluster. This value indicates the level of security that the cluster uses for communications between nodes in the cluster and also for communications between clusters.	no	no	only when changing from AUTHONLY or a cipher to EMPTY mode	no	for new connections
cnfsGrace The number of seconds a CNFS node denies new client requests after a node failover or failback.	yes	no	yes	no	immediately
cnfsMountdPort The port number to be used for rpc.mountd.	yes	no	no	no	immediately
cnfsNFSDprocs The number of nfsd kernel threads.	yes	no	no	no	if not immediately, on restart of the daemon
cnfsReboot Determines whether the node reboots when CNFS monitoring detects an unrecoverable problem.	yes	no	no	yes	immediately
cnfsSharedRoot Directory to be used by the clustered NFS subsystem.	yes	no	yes	no	immediately
cnfsVersions List of protocol versions that CNFS should start and monitor.	yes	no	yes	no	immediately
dataDiskCacheProtectionMethod Defines the cache protection method for disks that are used for the GPFS file system.	no	no	yes	no	on restart of the daemon
dataDiskWaitTimeForRecovery Controls the suspension of dataOnly disk recovery.	yes	no	no	yes	immediately
dataStructureDump Path for the storage of dumps.	yes	no	no	yes	if not immediately, on restart of the daemon

Table 3. Configuration attributes on the **mmchconfig** command (continued)

Attribute name and Description	-i option allowed	-I option allowed	GPFS must be stopped on all nodes	List of NodeNames allowed	Change takes effect
deadlockBreakupDelay When to attempt breaking up a detected deadlock.	yes	yes	no	no	immediately with -i or -I
deadlockDataCollectionDailyLimit Maximum number of times to collect debug data in 24 hours.	yes	yes	no	no	immediately with -i or -I
deadlockDataCollectionMinInterval Minimum interval between two consecutive collections of debug data.	yes	yes	no	no	immediately with -i or -I
deadlockDetectionThreshold Threshold for detecting deadlocks.	yes	yes	no	no	immediately with -i or -I
deadlockDetectionThresholdForShortWaiters Threshold for detecting deadlocks from short waiters.	yes	yes	no	no	immediately with -i or -I
deadlockDetectionThresholdIfOverloaded Threshold for detecting deadlocks when a cluster is overloaded.	yes	yes	no	no	immediately with -i or -I
deadlockOverloadThreshold Threshold for detecting cluster overload.	yes	yes	no	no	immediately with -i or -I
debugDataControl Controls the amount of debug data collected.	yes	no	no	yes	immediately
defaultMountDir Default parent directory for GPFS file systems.	yes	yes	no	no	for new file systems
disableInodeUpdateOnFdatasync Controls inode update on fdatasync for mtime and atime updates.	yes	yes	no	yes	immediately with -i or -I
dmapiDataEventRetry DMAPI attribute	yes	yes	no	yes	if not immediately, on restart of the daemon

Table 3. Configuration attributes on the **mmchconfig** command (continued)

Attribute name and Description	-i option allowed	-I option allowed	GPFS must be stopped on all nodes	List of NodeNames allowed	Change takes effect
dmapiEventTimeout DMAPI attribute	no	no	no	yes	on restart of the daemon
dmapiMountEvent DMAPI attribute	yes	yes	no	yes	if not immediately, on restart of the daemon
dmapiMountTimeout DMAPI attribute	yes	yes	no	yes	if not immediately, on restart of the daemon
dmapiSessionFailureTimeout DMAPI attribute	yes	yes	no	yes	if not immediately, on restart of the daemon
enableIPv6 Controls whether the GPFS daemon is to communicate through the IPv6 network.	no	no	only when enableIPv6 is set to yes	not applicable	if not immediately, on restart of the daemon
encryptionKeyCacheExpiration Specifies the refresh interval, in seconds, of the file system encryption key cache used internally by the mmfsd daemon.	no	no	no	yes	on restart of the daemon
enforceFilesetQuotaOnRoot Controls fileset quota settings for the root user.	yes	yes	no	no	if not immediately, on restart of the daemon
expelDataCollectionDailyLimit Maximum number of times to collect expel-related debug data in 24 hours.	yes	yes	no	no	immediately with -i or -I
expelDataCollectionMinInterval Minimum interval between two consecutive collections of expel-related debug data.	yes	yes	no	no	immediately with -i or -I
failureDetectionTime Indicates the amount of time it takes to detect that a node has failed.	no	no	yes	no	on restart of the daemon

Table 3. Configuration attributes on the **mmchconfig** command (continued)

Attribute name and Description	-i option allowed	-I option allowed	GPFS must be stopped on all nodes	List of NodeNames allowed	Change takes effect
fastestPolicyCmpThreshold Indicates the disk comparison count threshold, above which GPFS forces selection of this disk as the preferred disk to read.	yes	yes	no	yes	immediately with -i
fastestPolicyMaxValidPeriod Indicates the time period after which the disk's current evaluation is considered invalid.	yes	yes	no	yes	immediately with -i
fastestPolicyMinDiffPercent A percentage value indicating how GPFS selects the fastest between two disks.	yes	yes	no	yes	immediately with -i
fastestPolicyNumReadSamples Controls how many read samples taken to evaluate the disk's recent speed.	yes	yes	no	yes	immediately with -i
fileHeatLossPercent Specifies the reduction rate of FILE_HEAT value for every fileHeatPeriodMinutes of file inactivity.	yes	yes	no	no	if not immediately, on restart of the daemon
fileHeatPeriodMinutes Specifies the inactivity time before a file starts to lose FILE_HEAT value.	yes	yes	no	no	if not immediately, on restart of the daemon
FIPS1402mode Controls whether GPFS operates in FIPS 140-2 mode.	no	no	no	not applicable	on restart of the daemon
ignorePrefetchLUNCount The GPFS client node calculates the number of sequential access prefetch and write-behind threads to run concurrently for each file system by using the count of the number of LUNs in the file system and the value of maxMBps.	yes	yes	no	yes	immediately with -i
ignoreReplicationForQuota Specifies whether the quota commands ignore data replication factor.	yes	yes	no	no	immediately with -i or -I

Table 3. Configuration attributes on the **mmchconfig** command (continued)

Attribute name and Description	-i option allowed	-I option allowed	GPFS must be stopped on all nodes	List of NodeNames allowed	Change takes effect
ignoreReplicationOnStatfs Specifies if df command output on GPFS file system ignores data replication factor.	yes	yes	no	no	immediately with -i or -I
lrocData Controls whether user data is populated into the local read-only cache.	yes	yes	no	yes	immediately with -i or -I
lrocDataMaxFileSize Limits the data that may be saved in the local read-only cache to only the data from small files.	yes	yes	no	yes	immediately with -i or -I
lrocDataStubFileSize Limits the data that may be saved in the local read-only cache to only the data from the first portion of all files.	yes	yes	no	yes	immediately with -i or -I
lrocDirectories Controls whether directory blocks are populated into the local read-only cache.	yes	yes	no	yes	immediately with -i or -I
lrocEnableStoringClearText Controls whether encrypted file data can be read into a local read-only cache (LROC) device.	yes	yes	no	no	immediately with -i or -I
lrocInodes Controls whether inodes from open files are populated into the local read-only cache.	yes	yes	no	yes	immediately with -i or -I
maxblocksize Maximum file system block size allowed.	no	no	no	yes	on restart of the daemon
maxBufferDescs Can be tuned to cache very large files.	no	no	no	yes	on restart of the daemon
maxDownDisksForRecovery Maximum number of failed disks allowed for automatic recovery to continue.	yes	no	no	yes	immediately
maxFailedNodesForRecovery Maximum number of unavailable nodes allowed before automatic disk recovery is cancelled.	yes	no	no	yes	immediately

Table 3. Configuration attributes on the **mmchconfig** command (continued)

Attribute name and Description	-i option allowed	-I option allowed	GPFS must be stopped on all nodes	List of NodeNames allowed	Change takes effect
maxFcntlRangesPerFile Specifies the number of fctl locks that are allowed per file.	yes	yes	no	yes	if not immediately, on restart of the daemon
maxFilesToCache Number of inodes to cache for recently used files	no	no	no	yes	on restart of the daemon
maxMissedPingTimeout Handles high network latency in a short period of time.	no	no	no	no	on restart of the daemon
maxMBpS I/O throughput estimate	yes	yes	no	yes	if not immediately, on restart of the daemon
maxReceiverThreads Controls the maximum number of receiver threads which handle incoming TCP packets.	no	no	no	yes	on restart of the daemon
maxStatCache Number of inodes to keep in stat cache	no	no	no	yes	on restart of the daemon
metadataDiskWaitTimeForRecovery Controls the suspension of metadata disk recovery	yes	no	no	yes	immediately
minDiskWaitTimeForRecovery Controls the suspension of disk recovery	yes	no	no	yes	immediately
minMissedPingTimeout Handles high network latency in a short period of time	no	no	no	no	on restart of the daemon
mmapRangeLock Specifies POSIX or non-POSIX mmap byte-range semantics Note: The list of <i>NodeNames</i> is allowed, but it is not recommended.	yes	yes	no	yes	immediately
nfsPrefetchStrategy Optimizes prefetching for NFS file-style access patterns	yes	yes	no	yes	immediately with -i

Table 3. Configuration attributes on the **mmchconfig** command (continued)

Attribute name and Description	-i option allowed	-I option allowed	GPFS must be stopped on all nodes	List of NodeNames allowed	Change takes effect
nistCompliance Controls whether GPFS operates in NIST 800-131A mode for security transport mechanisms.	no	no	no	not applicable	if not immediately, on restart of the daemon
noSpaceEventInterval Time interval between noDiskSpace events of a file system	yes	yes	no	yes	if not immediately, on restart of the daemon
nsdBufSpace Percentage of the pagepool attribute that is reserved for the network transfer of NSD requests	yes	yes	no	yes	if not immediately, on restart of the daemon
nsdInlineWriteMax Specifies the maximum transaction size that can be sent as embedded data in an NSD-write RPC	yes	yes	no	yes	immediately with -i
nsdMaxWorkerThreads Sets the maximum number of NSD threads on an NSD server that concurrently transfers data with NSD clients	no	no	no	yes	on restart of the daemon
nsdMinWorkerThreads Used to increase the NSD server performance by providing a large number of dedicated threads for NSD service	no	no	no	yes	on restart of the daemon
nsdMultiQueue Sets the number of queues	yes	yes	no	yes	immediately with -i
nsdRAIDBufferSizePct Percentage of the page pool that is used for the IBM Storage Scale RAID vdisk buffer pool	yes	yes	no	yes	if not immediately, on restart of the daemon
nsdRAIDTracks Number of tracks in the IBM Storage Scale RAID buffer pool	yes	yes	no	yes	if not immediately, on restart of the daemon
nsdServerWaitTimeForMount Number of seconds to wait for an NSD server to come up	yes	yes	no	yes	if not immediately, on restart of the daemon

Table 3. Configuration attributes on the **mmchconfig** command (continued)

Attribute name and Description	-i option allowed	-I option allowed	GPFS must be stopped on all nodes	List of NodeNames allowed	Change takes effect
nsdServerWaitTimeWindowOnMount Time window to determine if quorum is to be considered <i>recently formed</i>	yes	yes	no	yes	if not immediately, on restart of the daemon
numaMemoryInterleave	no	no	no	yes	on restart of the daemon
pagepool Size of buffer cache on each node	yes	yes	no	yes	if not immediately, on restart of the daemon
pagepoolMaxPhysMemPct Percentage of physical memory that can be assigned to the page pool	no	no	no	yes	on restart of the daemon
pitWorkerThreadsPerNode Maximum number of threads to be involved in parallel processing on each node serving as a Parallel Inode Traversal (PIT) worker	yes	yes	no	yes	immediately with -i or -I
prefetchPct Acts as a guideline to limit the page pool space that is to be used for prefetch and write-behind buffers for active sequential streams	no	no	no	yes	on restart of the daemon
prefetchThreads Maximum number of threads dedicated to prefetching data	no	no	no	yes	on restart of the daemon
readReplicaPolicy The disk read replica policy	yes	yes	no	yes	immediately with -i
release=LATEST Complete the migration to a new release	yes	no	no	no	if not immediately, on restart of the daemon
restripeOnDiskFailure Specifies whether GPFS attempts to automatically recover from certain common disk failure situations.	yes	no	no	yes	immediately with -i
rpcPerfNumberDayIntervals Number of days that aggregated RPC data is saved	no	no	no	yes	on restart of the daemon

Table 3. Configuration attributes on the **mmchconfig** command (continued)

Attribute name and Description	-i option allowed	-I option allowed	GPFS must be stopped on all nodes	List of NodeNames allowed	Change takes effect
rpcPerfNumberHourIntervals Number of hours that aggregated RPC data is saved	no	no	no	yes	on restart of the daemon
rpcPerfNumberMinuteIntervals Number of minutes that aggregated RPC data is saved	no	no	no	yes	on restart of the daemon
rpcPerfNumberSecondIntervals Number of seconds that aggregated RPC data is saved	no	no	no	yes	on restart of the daemon
rpcPerfRawExecBufferSize The buffer size of the raw RPC execution times	no	no	no	yes	on restart of the daemon
rpcPerfRawStatBufferSize The buffer size of the raw RPC statistics	no	no	no	yes	on restart of the daemon
seqDiscardThreshold Detects a sequential read or write access pattern and specifies what has to be done with the page pool buffer after it is consumed or flushed by write-behind threads.	yes	yes	no	yes	immediately with -i
sidAutoMapRangeLength Controls the length of the reserved range for Windows SID to UNIX ID mapping	yes	yes	no	no	if not immediately, on restart of the daemon
sidAutoMapRangeStart Specifies the start of the reserved range for Windows SID to UNIX ID mapping	no	no	no	no	on restart of the daemon
syncbuffsperiteration Used to expedite buffer flush and the rename operations done by MapReduce jobs.	yes	yes	no	yes	immediately with -i
systemLogLevel Filters messages sent to the system log on Linux	yes	yes	no	yes	if not immediately, on restart of the daemon

Table 3. Configuration attributes on the **mmchconfig** command (continued)

Attribute name and Description	-i option allowed	-I option allowed	GPFS must be stopped on all nodes	List of NodeNames allowed	Change takes effect
subnets List of subnets to be used for most efficient daemon-to-daemon communication	no	no	no	yes	on restart of the daemon
sudoUser The default admin user name for logging on to nodes during the processing of an administration command. The GPFS daemon uses this user name only when sudo wrappers are enabled in the cluster and a program running at the root level invokes an administration command directly, without calling the sudo program.	yes	no	no	no	immediately
tiebreakerDisks (CCR repository) List of tiebreaker disks (NSDs)	no	no	no Note: If tiebreaker disks are part of the file system, GPFS must be up.	no	immediately
tiebreakerDisks (server-based repository) List of tiebreaker disks (NSDs)	no	no	yes	no	on restart of the daemon
uidDomain The UID domain name for the cluster.	no	no	yes	no	on restart of the daemon
unmountOnDiskFail Controls how the GPFS daemon responds when it detects a disk failure. For more information, see the topic <i>mmchconfig command</i> in the <i>IBM Storage Scale: Command and Programming Reference Guide</i> .	yes	yes	no	yes	Immediately
usePersistentReserve Enables or disables persistent reserve (PR) on the disks	no	no	yes	no	on restart of the daemon
verbsPorts Specifies InfiniBand device names, port numbers, and IP subnets.	no	no	no	yes	on restart of the daemon

Table 3. Configuration attributes on the **mmchconfig** command (continued)

Attribute name and Description	-i option allowed	-I option allowed	GPFS must be stopped on all nodes	List of NodeNames allowed	Change takes effect
verbsRdma Enables or disables InfiniBand RDMA using the Verbs API.	no	no	no	yes	on restart of the daemon
verbsRdmaCm Enables or disables InfiniBand RDMA_CM using the RDMA_CM API.	no	no	no	yes	on restart of the daemon
verbsRdmaRoCEToS Specifies the Type of Service (ToS) value for clusters using RDMA over Converged Ethernet (RoCE).	yes	yes	no	yes	if not immediately, on restart of the daemon
verbsRdmaSend Enables or disables the use of InfiniBand RDMA rather than TCP for most GPFS daemon-to-daemon communication.	no	no	no	yes	on restart of the daemon
verbsRecvBufferCount Defines the number of RDMA recv buffers created for each RDMA connection that is enabled for RDMA send when verbsRdmaSend is enabled.	no	no	no	yes	on restart of the daemon
verbsRecvBufferSize Defines the size, in bytes, of the RDMA send and recv buffers used for RDMA connections that are enabled for RDMA send when verbsRdmaSend is enabled.	no	no	no	yes	on restart of the daemon
workerThreads Sets an integrated group of variables that tune file system performance.	no	no	no	yes	on restart of the daemon
worker1Threads Sets the maximum number of concurrent file operations	yes (only when adjusting value down)	yes (only when adjusting value down)	no	yes	on restart of the daemon
writebehindThreshold Specifies the point at which GPFS starts flushing new data out of the page pool for a file that is being written sequentially.	yes	yes	no	yes	immediately with -i

Specify the nodes you want to target for change and the attributes with their new values on the **mmchconfig** command. For example, to change the pagepool value for each node in the GPFS cluster immediately, enter:

```
mmchconfig pagepool=100M -i
```

The system displays information similar to:

```
mmchconfig: Command successfully completed
mmchconfig: 6027-1371 Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

For more information, see **mmchconfig command** in *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

[Displaying cluster configuration information](#)

Use the **mmlscluster** command to display cluster configuration information.

[Security mode](#)

The security mode of a cluster determines the level of security that the cluster provides for communications between nodes in the cluster and also for communications between clusters.

[Minimum release level of a cluster](#)

The minimum release level of a cluster is the currently enabled level of functionality of the cluster. It is expressed as an IBM Storage Scale version number, such as 5.1.7.

[Cluster quorum with quorum nodes](#)

Cluster quorum defines the minimum number of GPFS quorum nodes that must have the GPFS daemon actively running on them for the cluster to be operational. This number is half of the defined GPFS quorum nodes plus one, so it is a good idea to have an odd number of GPFS quorum nodes defined. Cluster quorum using only GPFS quorum nodes is the default quorum algorithm.

[Cluster quorum with quorum nodes and tiebreaker disks](#)

To use cluster quorum with quorum nodes and tiebreaker disks, more requirements apply.

[Configuring cluster configuration repository](#)

The cluster configuration repository (CCR) is an internal configuration store for various IBM Storage Scale components. It is not meant to be used directly by the customer.

Related tasks

[Creating your GPFS cluster](#)

You must first create a GPFS cluster by issuing the **mmcrccluster** command.

[Adding nodes to a GPFS cluster](#)

You can add nodes to an existing GPFS cluster by issuing the **mmaddnode** command or by using the IBM Storage Scale installation toolkit. The new nodes are available immediately after the successful completion of the process.

[Deleting nodes from a GPFS cluster](#)

You can delete nodes from a GPFS cluster by issuing the **mmde1node** command.

[Running IBM Storage Scale commands without remote root login](#)

With sudo wrapper scripts you can avoid configuring nodes to allow remote root login.

[Displaying and changing the file system manager node](#)

In general, GPFS performs the same functions on all nodes. There are also cases where one node provides a more global function that affects the operation of multiple nodes. For example, each file system is assigned a node that functions as a file system manager.

[Starting and stopping GPFS](#)

You can use the **mmstartup** and **mmshutdown** commands to start and stop GPFS on new or existing clusters.

[Shutting down an IBM Storage Scale cluster](#)

You might need to shut down an IBM Storage Scale cluster in an emergency.

Security mode

The security mode of a cluster determines the level of security that the cluster provides for communications between nodes in the cluster and also for communications between clusters.

There are three security modes:

EMPTY

The receiving node and the sending node do not authenticate each other, do not encrypt transmitted data, and do not check the integrity of transmitted data.

AUTHONLY

The sending and receiving nodes authenticate each other with a TLS handshake and then close the TLS connection. Communication continues in the clear. The nodes do not encrypt transmitted data and do not check data integrity.

Cipher

To set this mode, you must specify the name of a supported cipher, such as AES128-GCM-SHA256. The sending and receiving nodes authenticate each other with a TLS handshake. A TLS connection is established. The transmitted data is encrypted with the specified cipher and is checked for data integrity.

To find a list of supported ciphers, choose one of the following methods:

- See the frequently answered questions (FAQs) in [IBM Storage Scale FAQ in IBM Documentation](#).
- Enter the following command at the command line:

```
mmauth show ciphers
```

For FIPS 140-2 considerations, see the *Encryption* topic in the *IBM Storage Scale: Administration Guide*.

For both the AUTHONLY mode and the cipher mode, the cluster automatically generates a public or private key pair when the mode is set. However, for communication between clusters, the system administrators are still responsible for exchanging public keys.

In IBM Storage Scale, the default security mode is AUTHONLY. The **mmcrcluster** command sets the mode when it creates the cluster. You can display the security mode by running the following command:

```
mmlsconfig cipherlist
```

You can change the security mode with the following command:

```
mmchconfig cipherlist=security_mode
```

If you are changing the security mode from EMPTY to another mode, you can do so without stopping the GPFS daemon. However, if you are changing the security mode from another mode to EMPTY, you must stop the GPFS daemon on all the nodes in the cluster. Change the security mode to EMPTY and then restart the GPFS daemon.

The default security mode is EMPTY in IBM Storage Scale 4.1 or earlier and is AUTHONLY in IBM Storage Scale 4.2 or later. If you migrate a cluster from IBM Storage Scale 4.1 to 4.2 or later by running **mmchconfig release=LATEST**, the command checks the security mode. If the mode is EMPTY, the command fails with an error message. You then can do either of two actions:

- Change the security mode to a valid value other than EMPTY, such as AUTHONLY, and rerun the **mmchconfig release=LATEST** command.
- Leave the security mode set to EMPTY and re-run the **mmchconfig release=LATEST** command with the option **--accept-empty-cipherlist-security**.

For more information, see *Completing the migration to a new level of IBM Storage Scale* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Configuring the security mode to a setting other than EMPTY (that is, either AUTHONLY or a supported cipher) requires the use of the GSKit toolkit for encryption and authentication. As such, the `gpfs.gskit` package, which is available on all editions, should be installed.

Related concepts

[Displaying cluster configuration information](#)

Use the `mmIscluster` command to display cluster configuration information.

[Minimum release level of a cluster](#)

The minimum release level of a cluster is the currently enabled level of functionality of the cluster. It is expressed as an IBM Storage Scale version number, such as 5.1.7.

[Cluster quorum with quorum nodes](#)

Cluster quorum defines the minimum number of GPFS quorum nodes that must have the GPFS daemon actively running on them for the cluster to be operational. This number is half of the defined GPFS quorum nodes plus one, so it is a good idea to have an odd number of GPFS quorum nodes defined. Cluster quorum using only GPFS quorum nodes is the default quorum algorithm.

[Cluster quorum with quorum nodes and tiebreaker disks](#)

To use cluster quorum with quorum nodes and tiebreaker disks, more requirements apply.

[Configuring cluster configuration repository](#)

The cluster configuration repository (CCR) is an internal configuration store for various IBM Storage Scale components. It is not meant to be used directly by the customer.

Related tasks

[Creating your GPFS cluster](#)

You must first create a GPFS cluster by issuing the `mmcrccluster` command.

[Adding nodes to a GPFS cluster](#)

You can add nodes to an existing GPFS cluster by issuing the `mmAddnode` command or by using the IBM Storage Scale installation toolkit. The new nodes are available immediately after the successful completion of the process.

[Deleting nodes from a GPFS cluster](#)

You can delete nodes from a GPFS cluster by issuing the `mmDelnode` command.

[Changing the GPFS cluster configuration data](#)

You can use the `mmchcluster` or `mmchconfig` command to change the configuration attributes.

[Running IBM Storage Scale commands without remote root login](#)

With sudo wrapper scripts you can avoid configuring nodes to allow remote root login.

[Displaying and changing the file system manager node](#)

In general, GPFS performs the same functions on all nodes. There are also cases where one node provides a more global function that affects the operation of multiple nodes. For example, each file system is assigned a node that functions as a file system manager.

[Starting and stopping GPFS](#)

You can use the `mmStartup` and `mmShutdown` commands to start and stop GPFS on new or existing clusters.

[Shutting down an IBM Storage Scale cluster](#)

You might need to shut down an IBM Storage Scale cluster in an emergency.

Setting the security mode for internode communications in a cluster

IBM Storage Scale supports the secure data transit for internode communications within a single cluster.

Enable encryption of the data over wire for the internode communications between the IBM Storage Scale systems by using the following setting:

1. Set the `cipherList` value to one of the supported ciphers by using the `mmchconfig` command.

For example,

```
# mmchconfig cipherList=AES256-SHA256
```

By setting the **cipherList** value, the data that is exchanged between the nodes in a single cluster of IBM Storage Scale is encrypted with the AES256-SHA256 cipher.

2. Restart the GPFS daemon across the cluster so that the security setting is in effect.

Important: To keep cluster services operational, you can start the daemons in a rolling fashion, one node at a time. The new security mode takes effect for each new TCP connection that is established. After the daemons on all nodes in a cluster are restarted, the security mode takes effect for all TCP connections.

The **cipherList** setting does not affect the existing TCP connections. These TCP connections remain in their previous setting, which is likely to be the *AUTHONLY* mode.

Note: TCP connections that are established for the clustered configuration repository (CCR) operate in the *AUTHONLY* mode.

Minimum release level of a cluster

The minimum release level of a cluster is the currently enabled level of functionality of the cluster. It is expressed as an IBM Storage Scale version number, such as 5.1.7.

[“Minimum release level” on page 24](#)

[“Access to files in remote clusters” on page 25](#)

Minimum release level

The minimum release level is a common level of functionality that all the nodes of a cluster can participate in. For example, if the minimum release level of a cluster is 5.1.7, the nodes in the cluster can use all the features that are installed with IBM Storage Scale version 5.1.7 or earlier.

Note: The minimum release level is a cluster-level attribute. It does not apply across clusters. Each cluster has its own minimum release level.

To maintain a common level of functionality in the cluster, IBM Storage Scale does not enable installed features that require a version of IBM Storage Scale that is later than the minimum release level. For example, if some or even all of the nodes of a cluster are installed with version 5.0.2.0 of IBM Storage Scale, but the minimum release level is 5.0.0.0, nodes in the cluster cannot use installed features that require a later version of the product than 5.0.0.0. For instance, nodes in the cluster cannot use the file audit logging feature, which was introduced in 5.0.2 and requires version 5.0.2 or later. File audit logging is not enabled in the cluster until the minimum release level is raised to 5.0.2 or later.

To display the minimum release level of a cluster, issue the **mmlsconfig** command:

```
# mmlsconfig
Configuration data for cluster example.cluster:
-----
...
minReleaseLevel 5.0.0.0
...
```

When a cluster is created, the minimum release level of the cluster is set to the release level of the node where the **mmcrcluster** command is issued.

Tip: The results of running the **mmcrcluster** command depend partly on the relative release levels of the nodes that you are including in the new cluster:

- If the **mmcrcluster** command is issued on the node with the lowest release level in the cluster, then cluster creation succeeds and the minimum release level of the cluster corresponds to the release level of the node from which the cluster was created.
- If the **mmcrcluster** command is issued on a node other than the node with the lowest release level, then one of two outcomes can occur. Cluster creation might fail, or it might succeed but exclude nodes

with lower release levels from the cluster. In either case the command might display an error message like the following one:

```
6027-1599 The level of GPFS on node vmip135.gpfs.net does not support the requested action.
```

Important: Nodes in a cluster can run different versions of IBM Storage Scale only if the versions are compatible. For more information, see the subsection "Can different IBM Storage Scale maintenance levels coexist?" in [IBM Storage Scale FAQ in IBM Documentation](#).

To increase the minimum release level to the latest common level of functionality, issue the following command:

```
mmchconfig release=LATEST
```

For more information, see the topic *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide*. Issuing the `mmchconfig release=LATEST` command is frequently one of the final steps in upgrading a cluster to a later version of IBM Storage Scale. For more information, see the topic *Completing the upgrade to a new level of IBM Storage Scale* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.



Warning: You cannot decrease the minimum release level of a cluster or revert it to the previous level, except by a lengthy process of uninstalling and reinstalling. For more information, see the topic *Reverting to the previous level of IBM Storage Scale* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

When you raise the minimum release level by running the `mmchconfig release=LATEST` command, nodes that are installed with earlier versions of IBM Storage Scale cannot be added to the cluster unless they are upgraded to a version greater than or equal to the minimum release level of the cluster.

Some features of IBM Storage Scale that are available at an earlier minimum release level might work differently or not be available at a later minimum release level. Examples:

1. In the `mmcrfs` command and the `mmchfs` command, the default value of the `-S` parameter can be different depending on the minimum release level. For more information, see the topics *mmcrfs command* and *mmchfs command* in the *IBM Storage Scale: Command and Programming Reference Guide*.
2. The default value of the `CipherList` parameter of the `mmauth` command and the `cipherList` parameter of the `mmchconfig` command can be different depending on the minimum release level. For more information, see the topic *Security mode* and the topic *Completing the upgrade to a new level of IBM Storage Scale* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Access to files in remote clusters

Nodes in one cluster can mount file systems in another cluster if the following requirements are met:

- The minimum release levels of the two clusters are compatible. For more information, see the *Can different IBM Storage Scale™ maintenance levels coexist?* in [IBM Storage Scale FAQ in IBM Documentation](#).
- The version of the accessing node is greater than or equal to the file system format version of the file system being mounted.

For more information, see [Chapter 27, “File system format changes between versions of IBM Storage Scale,” on page 267](#) and the description of the `--version` parameter in [the `mmcrfs` command](#) in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

[Displaying cluster configuration information](#)

Use the `mmlscluster` command to display cluster configuration information.

[Security mode](#)

The security mode of a cluster determines the level of security that the cluster provides for communications between nodes in the cluster and also for communications between clusters.

Cluster quorum with quorum nodes

Cluster quorum defines the minimum number of GPFS quorum nodes that must have the GPFS daemon actively running on them for the cluster to be operational. This number is half of the defined GPFS quorum nodes plus one, so it is a good idea to have an odd number of GPFS quorum nodes defined. Cluster quorum using only GPFS quorum nodes is the default quorum algorithm.

Cluster quorum with quorum nodes and tiebreaker disks

To use cluster quorum with quorum nodes and tiebreaker disks, more requirements apply.

Configuring cluster configuration repository

The cluster configuration repository (CCR) is an internal configuration store for various IBM Storage Scale components. It is not meant to be used directly by the customer.

Related tasks

Creating your GPFS cluster

You must first create a GPFS cluster by issuing the **mmcrcluster** command.

Adding nodes to a GPFS cluster

You can add nodes to an existing GPFS cluster by issuing the **mmaddnode** command or by using the IBM Storage Scale installation toolkit. The new nodes are available immediately after the successful completion of the process.

Deleting nodes from a GPFS cluster

You can delete nodes from a GPFS cluster by issuing the **mmdelnode** command.

Changing the GPFS cluster configuration data

You can use the **mmchcluster** or **mmchconfig** command to change the configuration attributes.

Running IBM Storage Scale commands without remote root login

With sudo wrapper scripts you can avoid configuring nodes to allow remote root login.

Displaying and changing the file system manager node

In general, GPFS performs the same functions on all nodes. There are also cases where one node provides a more global function that affects the operation of multiple nodes. For example, each file system is assigned a node that functions as a file system manager.

Starting and stopping GPFS

You can use the **mmstartup** and **mmshutdown** commands to start and stop GPFS on new or existing clusters.

Shutting down an IBM Storage Scale cluster

You might need to shut down an IBM Storage Scale cluster in an emergency.

Running IBM Storage Scale commands without remote root login

With sudo wrapper scripts you can avoid configuring nodes to allow remote root login.

Every administration node in the IBM Storage Scale cluster must be able to run administration commands on any other node in the cluster. Each administration node must be able to do so without the use of a password and without producing any extraneous messages. Also, most of the IBM Storage Scale administration commands must run at the root level. One solution to meet these requirements is to configure each node to permit general remote login to its root user ID. However, there are secure solutions available that do not require root-level login.

You can use the sudo program to eliminate direct root login. With sudo wrapper, you can launch IBM Storage Scale administration commands with a sudo wrapper script. This script uses ssh to log in to the remote node using a non-root ID, and then invokes the sudo program on the remote node to run the commands with root-level privileges. The root user on an administration node still needs to be able to log in to all nodes in the cluster as the non-root ID, without being prompted for a password.

Note: Only the instance of sudo that is shipped natively with the Linux operating system or included in the AIX toolbox is supported. Other sudo-like frameworks might only be supported after a technical

compatibility review by IBM. Ask your sales representative to contact IBM Storage Scale development about the RPQ or SCORE process.

Note:

- Sudo wrappers are not supported on clusters where one or more of the nodes is running the Windows operating system.
- Sudo wrappers are not supported with clustered NFS (cNFS).
- The installation toolkit is not supported in a sudo wrapper environment.

To use sudo wrappers, complete the tasks in the following topics:

Related concepts

[Displaying cluster configuration information](#)

Use the **mm1scluster** command to display cluster configuration information.

[Security mode](#)

The security mode of a cluster determines the level of security that the cluster provides for communications between nodes in the cluster and also for communications between clusters.

[Minimum release level of a cluster](#)

The minimum release level of a cluster is the currently enabled level of functionality of the cluster. It is expressed as an IBM Storage Scale version number, such as 5.1.7.

[Cluster quorum with quorum nodes](#)

Cluster quorum defines the minimum number of GPFS quorum nodes that must have the GPFS daemon actively running on them for the cluster to be operational. This number is half of the defined GPFS quorum nodes plus one, so it is a good idea to have an odd number of GPFS quorum nodes defined. Cluster quorum using only GPFS quorum nodes is the default quorum algorithm.

[Cluster quorum with quorum nodes and tiebreaker disks](#)

To use cluster quorum with quorum nodes and tiebreaker disks, more requirements apply.

[Configuring cluster configuration repository](#)

The cluster configuration repository (CCR) is an internal configuration store for various IBM Storage Scale components. It is not meant to be used directly by the customer.

Related tasks

[Creating your GPFS cluster](#)

You must first create a GPFS cluster by issuing the **mmcrccluster** command.

[Adding nodes to a GPFS cluster](#)

You can add nodes to an existing GPFS cluster by issuing the **mmaddnode** command or by using the IBM Storage Scale installation toolkit. The new nodes are available immediately after the successful completion of the process.

[Deleting nodes from a GPFS cluster](#)

You can delete nodes from a GPFS cluster by issuing the **mmdeinode** command.

[Changing the GPFS cluster configuration data](#)

You can use the **mmchcluster** or **mmchconfig** command to change the configuration attributes.

[Displaying and changing the file system manager node](#)

In general, GPFS performs the same functions on all nodes. There are also cases where one node provides a more global function that affects the operation of multiple nodes. For example, each file system is assigned a node that functions as a file system manager.

[Starting and stopping GPFS](#)

You can use the **mmstartup** and **mmshutdown** commands to start and stop GPFS on new or existing clusters.

[Shutting down an IBM Storage Scale cluster](#)

You might need to shut down an IBM Storage Scale cluster in an emergency.

Configuring sudo

The system administrator must configure sudo by modifying the `sudoers` file. IBM Storage Scale installs a sample of the modified `sudoers` file as `/usr/lpp/mmfs/samples/sudoers.sample`.

Do the following steps before you configure sudo:

1. Create a user and group to run administration commands.

Note: The examples in this section have the user name `gpfsadmin` and the group `gpfs`.

2. Allow the root user from an administration node to run commands on all nodes including the current node with user ID `gpfsadmin` without being prompted for a password. For example, the root user must be able to issue a command like the following one without being prompted for a password:

```
# ssh c6f2bc4n8 -l gpfsadmin /bin/whoami  
gpfsadmin
```

3. Install the sudo program. Sudo is free open source software that is distributed under a license.

Do the following steps on each node in the cluster:

1. Open the `/etc/sudoers` file with a text editor. The sudo installation includes the `visudo` editor, which checks the syntax of the file before closing.
2. Add the following commands to the file. **Important:** Enter each command on a single line:

```
# Preserve GPFS environment variables:  
Defaults env_keep += "MMMODE environmentType GPFS_rshPath GPFS_rcpPath mmScriptTrace GPFSCMDPORTRANGE GPFS_CIM_MSG_FORMAT"  
  
# Allow members of the gpfs group to run all commands but only selected commands without a password:  
%gpfs ALL=(ALL) NOPASSWD: ALL, NOPASSWD: /usr/lpp/mmfs/bin/mmremote  
  
# Disable requiretty for group gpfs:  
Defaults:%gpfs !requiretty
```

The first line preserves the environment variables that the IBM Storage Scale administration commands need to run. The second line allows the users in the `gpfs` group to run administration commands without being prompted for a password. The third line disables `requiretty`. When this flag is enabled, sudo blocks the commands that do not originate from a TTY session.

Note: As of IBM Storage Scale 5.1.0, you no longer need to add commands such as `scp`, `echo`, and `mmsdrrestore` to the `sudoers` file.

If the minimum release level of the cluster is earlier than 5.1.0, add the following command to the file as the second line:

```
# Allow members of the gpfs group to run all commands but only selected commands without a  
password:  
%gpfs ALL=(ALL) NOPASSWD: ALL, NOPASSWD: /usr/lpp/mmfs/bin/mmremote, /usr/bin/scp, /bin/  
echo, /usr/lpp/mmfs/bin/mmsdrrestore
```

3. Perform the following steps to verify that the `sshwrap` and `scpwrap` scripts work correctly.

- a) `sshwrap` is an IBM Storage Scale sudo wrapper script for the remote shell command that is installed with IBM Storage Scale. To verify that it works correctly, run the following command as the `gpfsadmin` user:

```
sudo /usr/lpp/mmfs/bin/mmcommon test sshwrap nodeName  
[sudo] password for gpfsadmin:  
mmcommon test sshwrap: Command successfully completed
```

Note: Here `nodeName` is the name of an IBM Storage Scale node in the cluster.

- b) `scpwrap` is an IBM Storage Scale sudo wrapper script for the remote file copy command that is installed with IBM Storage Scale. To verify that it works correctly, run the following command as the `gpfsadmin` user:

```
sudo /usr/lpp/mmfs/bin/mmcommon test scpwrap nodeName  
mmcommon test scpwrap: Command successfully completed
```

Note: Here nodeName is the name of an IBM Storage Scale node in the cluster.

Sudo is now configured to run administration commands without remote root login.

Configuring the cluster to use sudo wrapper scripts

The system administrator must configure the IBM Storage Scale cluster to call the sudo wrapper scripts sshwrap and scpwrap to run IBM Storage Scale administration commands. To configure the cluster, either run the **mmcrccluster** command or the **mmchcluster** command with the **--use-sudo-wrapper** option.

Ensure that the **--sudo-user UserName** option is set as required for the IBM Storage Scale GUI and call home components when you use the **mmcrccluster** command or the **mmchcluster** command to configure the sudo wrappers.

Follow these steps to configure a new cluster or an existing cluster to call the sudo wrapper scripts:

- To configure a new cluster to call the sudo wrapper scripts, use these steps:
 - Log in with the user ID. This example uses gpfsadmin as the user ID.
 - Issue the **mmcrccluster** command with the **--use-sudo-wrapper** option as shown in the following example:

```
$ sudo /usr/lpp/mmfs/bin/mmcrccluster --use-sudo-wrapper -N c13c1apv7:quorum,c13c1apv8  
mmcrccluster: Performing preliminary node verification ...  
mmcrccluster: Processing quorum and other critical nodes ...  
mmcrccluster: Processing the rest of the nodes ...  
mmcrccluster: Finalizing the cluster data structures ...  
mmcrccluster: Command successfully completed mmcrccluster:  
Warning: Not all nodes have proper GPFS license designations.  
Use the mmchlicense command to designate licenses as needed.  
mmcrccluster: Propagating the cluster configuration data to all  
affected nodes. This is an asynchronous process
```

- To verify that the cluster is using sudo wrappers, run the **mmclscluster** command as shown in the following example:

```
gpfsadmin@c13c1apv7 admin]$mmclscluster  
GPFS cluster information  
=====  
GPFS cluster name: c13c1apv7.gpfs.net  
GPFS cluster id: 12275146245716580740  
GPFS UID domain: c13c1apv7.gpfs.net  
Remote shell command: /usr/lpp/mmfs/bin/sshwrap  
Remote file copy command: sudo wrapper in use  
Repository type: CCR  
Node Daemon node name IP address Admin node name Designation  
-----  
1 c13c1apv7.gpfs.net 192.168.148.117 c13c1apv7.gpfs.net quorum  
2 c13c1apv8.gpfs.net 192.168.148.118 c13c1apv8.gpfs.net
```

- To configure an existing cluster to call the sudo wrapper scripts, use these steps:
 - Log in with the user ID. This example uses gpfsadmin as the user ID.
 - Issue the **mmchcluster** command with the **--use-sudo-wrapper** option to use the sudo wrappers:

```
sudo /usr/lpp/mmfs/bin/mmchcluster --use-sudo-wrapper
```

- To verify that the cluster is using sudo wrappers, run the **mmclscluster** command with no parameters. If the sudo wrapper is configured properly, the output must contain the following two lines:

```
Remote shell command: sudo wrapper in use  
Remote file copy command: sudo wrapper in use
```

Configuring IBM Storage Scale GUI to use sudo wrapper

The GUI can be configured to run on a cluster where remote root access is disabled and sudo wrappers are used. On such a cluster, the GUI process still runs as root but it issues ssh to other nodes using a user name for which sudo wrappers were configured.

Make the following configuration change to use the IBM Storage Scale management GUI on a cluster where sudo wrappers are used:

Issue the **systemctl restart gpfs gui** command to restart the GUI.

This ensures that the GUI is initialized with the new value of **sudoUser**, as specified in the **mmcrcluster** or **mmchcluster** command.

Passwordless ssh is set up between the root user on the node where the GUI is running on all the remote nodes in the cluster. The ssh calls are equivalent to *ssh gpfsadmin@destination-node*. Therefore, it is not necessary to set up passwordless ssh between *gpfsadmin* users on any two nodes. The root user of the node where the GUI is running can do passwordless ssh to any other node using the *gpfsadmin* user login. So, unidirectional access from the GUI node to the remote nodes as *gpfsadmin* user is enough.

Note: If sudo wrappers are enabled on the cluster but GUI is not configured for it, the system raises an event.

Configuring a cluster to stop using sudo wrapper scripts

Follow these directions to stop using sudo wrapper scripts in the IBM Storage Scale cluster.

To stop using sudo wrappers, run the **mmchcluster** command with the **--nouse-sudo-wrapper** option as shown in the following example:

```
$sudo /usr/lpp/mmfs/bin/mmchcluster --nouse-sudo-wrapper
```

The cluster stops calling the sudo wrapper scripts to run the remote administration commands.

Root-level processes that call administration commands directly

With the **sudoUser** attribute, you can enable root-level background processes to call administration commands directly while sudo wrappers are enabled.

When sudo wrappers are enabled and a root-level background process calls an administration command directly rather than through **sudo**, the administration command typically fails. Examples of such a root-level process are the **cron** program and IBM Storage Scale callback programs. Such processes call administration commands directly even when sudo wrappers are enabled.

In the failing scenario, the GPFS daemon that processes the administration command encounters a login error when it tries to run an internal command on another node as the root user. When sudo wrappers are enabled, nodes typically do not allow root-level logins by other nodes. (That is the advantage of having sudo wrappers.) When the root-level login fails, the GPFS daemon that is processing the administration command cannot complete the command and returns an error.

To avoid this problem, you can set the **sudoUser** attribute to a non-root admin user ID that can log in to any node in the cluster without being prompted for a password. You can specify the same admin user ID that you used to configure **sudo**. For more information on the admin user ID, see [“Configuring sudo” on page 28](#).

You can set the **sudoUser** attribute in the **mmchconfig** command (the **sudoUser** attribute), **mmcrcluster** command (the **--sudo-user** parameter), or **mmcrcluster** command (the **--sudo-user** parameter) as described in the *IBM Storage Scale: Command and Programming Reference Guide*.

Cluster quorum with quorum nodes

Cluster quorum defines the minimum number of GPFS quorum nodes that must have the GPFS daemon actively running on them for the cluster to be operational. This number is half of the defined GPFS quorum

nodes plus one, so it is a good idea to have an odd number of GPFS quorum nodes defined. Cluster quorum using only GPFS quorum nodes is the default quorum algorithm.

For more information on node quorum, see the *Quorum* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Related concepts

[Displaying cluster configuration information](#)

Use the **mm1scluster** command to display cluster configuration information.

[Security mode](#)

The security mode of a cluster determines the level of security that the cluster provides for communications between nodes in the cluster and also for communications between clusters.

[Minimum release level of a cluster](#)

The minimum release level of a cluster is the currently enabled level of functionality of the cluster. It is expressed as an IBM Storage Scale version number, such as 5.1.7.

[Cluster quorum with quorum nodes and tiebreaker disks](#)

To use cluster quorum with quorum nodes and tiebreaker disks, more requirements apply.

[Configuring cluster configuration repository](#)

The cluster configuration repository (CCR) is an internal configuration store for various IBM Storage Scale components. It is not meant to be used directly by the customer.

Related tasks

[Creating your GPFS cluster](#)

You must first create a GPFS cluster by issuing the **mmcrccluster** command.

[Adding nodes to a GPFS cluster](#)

You can add nodes to an existing GPFS cluster by issuing the **mmaddnode** command or by using the IBM Storage Scale installation toolkit. The new nodes are available immediately after the successful completion of the process.

[Deleting nodes from a GPFS cluster](#)

You can delete nodes from a GPFS cluster by issuing the **mmdeinode** command.

[Changing the GPFS cluster configuration data](#)

You can use the **mmchcluster** or **mmchconfig** command to change the configuration attributes.

[Running IBM Storage Scale commands without remote root login](#)

With sudo wrapper scripts you can avoid configuring nodes to allow remote root login.

[Displaying and changing the file system manager node](#)

In general, GPFS performs the same functions on all nodes. There are also cases where one node provides a more global function that affects the operation of multiple nodes. For example, each file system is assigned a node that functions as a file system manager.

[Starting and stopping GPFS](#)

You can use the **mmstartup** and **mmshutdown** commands to start and stop GPFS on new or existing clusters.

[Shutting down an IBM Storage Scale cluster](#)

You might need to shut down an IBM Storage Scale cluster in an emergency.

Cluster quorum with quorum nodes and tiebreaker disks

To use cluster quorum with quorum nodes and tiebreaker disks, more requirements apply.

If a cluster is configured with quorum nodes and tiebreaker disks, then it can operate with as few as one available quorum node. This condition is applicable only when the quorum node has access to most of the tiebreaker disks. Setting up this configuration includes the following general steps:

1. Designate one or more nodes as quorum nodes.
2. Define one to three disks as tiebreaker disks, by issuing the **mmchconfig** command with the **tiebreakerDisks** parameter. You can designate any disk to be a tiebreaker disk.

For more information, see the *Quorum* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

For example, to add tiebreaker disks to a cluster that is configured only with quorum nodes (without tiebreaker disks) issue the following command:

```
mmchconfig tiebreakerDisks="nsdName[;nsdName...]"
```

where *nsdName* is the name of a disk.

The following requirements must be met:

- If a GPFS cluster includes more than eight quorum nodes, the cluster configuration cannot be changed to cluster quorum with quorum nodes and tiebreaker disks.
- The cluster can have up to three tiebreaker disks.
- The tiebreaker disks must be directly attached to all the quorum nodes.
- When you add tiebreaker disks, you might have to shut down the GPFS daemon:
 - If the tiebreaker disks are part of a file system, then the GPFS daemon must be up and running.
 - If the tiebreaker disks are not part of a file system, then the GPFS daemon is either running or shut down.
- If a cluster uses the traditional server-based (non-CCR) configuration repository, you must shut down the GPFS daemons on all the nodes in the cluster before you issue the **mmchconfig tiebreakerDisks** command.

To revert from cluster quorum with quorum nodes and tiebreaker disks to cluster quorum with quorum nodes only, issue the following command:

```
mmchconfig tiebreakerDisks=DEFAULT
```

Related concepts

[Displaying cluster configuration information](#)

Use the **mmlscluster** command to display cluster configuration information.

[Security mode](#)

The security mode of a cluster determines the level of security that the cluster provides for communications between nodes in the cluster and also for communications between clusters.

[Minimum release level of a cluster](#)

The minimum release level of a cluster is the currently enabled level of functionality of the cluster. It is expressed as an IBM Storage Scale version number, such as 5.1.7.

[Cluster quorum with quorum nodes](#)

Cluster quorum defines the minimum number of GPFS quorum nodes that must have the GPFS daemon actively running on them for the cluster to be operational. This number is half of the defined GPFS quorum nodes plus one, so it is a good idea to have an odd number of GPFS quorum nodes defined. Cluster quorum using only GPFS quorum nodes is the default quorum algorithm.

[Configuring cluster configuration repository](#)

The cluster configuration repository (CCR) is an internal configuration store for various IBM Storage Scale components. It is not meant to be used directly by the customer.

Related tasks

[Creating your GPFS cluster](#)

You must first create a GPFS cluster by issuing the **mmcrcluster** command.

[Adding nodes to a GPFS cluster](#)

You can add nodes to an existing GPFS cluster by issuing the **mmaddnode** command or by using the IBM Storage Scale installation toolkit. The new nodes are available immediately after the successful completion of the process.

[Deleting nodes from a GPFS cluster](#)

You can delete nodes from a GPFS cluster by issuing the **mmdeinode** command.

Changing the GPFS cluster configuration data

You can use the **mmchcluster** or **mmchconfig** command to change the configuration attributes.

Running IBM Storage Scale commands without remote root login

With sudo wrapper scripts you can avoid configuring nodes to allow remote root login.

Displaying and changing the file system manager node

In general, GPFS performs the same functions on all nodes. There are also cases where one node provides a more global function that affects the operation of multiple nodes. For example, each file system is assigned a node that functions as a file system manager.

Starting and stopping GPFS

You can use the **mmstartup** and **mmshutdown** commands to start and stop GPFS on new or existing clusters.

Shutting down an IBM Storage Scale cluster

You might need to shut down an IBM Storage Scale cluster in an emergency.

Displaying and changing the file system manager node

In general, GPFS performs the same functions on all nodes. There are also cases where one node provides a more global function that affects the operation of multiple nodes. For example, each file system is assigned a node that functions as a file system manager.

For more information, see *Special management functions* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

The node that is the file system manager can also be used for applications. In some cases involving very large clusters or applications that place a high stress on metadata operations, it may be useful to specify which nodes are used as file system managers. Applications that place a high stress on metadata operations are usually those that involve large numbers of very small files, or that do very fine-grain parallel write-sharing among multiple nodes.

You can display the file system manager node by issuing the **mmlsmgr** command. You can display the information for an individual file system, a list of file systems, or for all of the file systems in the cluster. For example, to display the file system manager for the file system **fs1**, enter:

```
mmlsmgr fs1
```

The output shows the device name of the file system and the file system manager's node number and name:

```
file system      manager node      [from 19.134.68.69 (k164n05)]
-----
fs1            19.134.68.70 (k164n06)
```

For more information, see ***mmlsmgr command*** in *IBM Storage Scale: Command and Programming Reference Guide*.

You can change the file system manager node for an individual file system by issuing the **mmchmgr** command. For example, to change the file system manager node for the file system **fs1** to **k145n32**, enter:

```
mmchmgr fs1 k145n32
```

The output shows the file system manager's node number and name, in parentheses, as recorded in the GPFS cluster data:

```
GPFS: 6027-628 Sending migrate request to current manager node 19.134.68.69 (k145n30).
GPFS: 6027-629 [N] Node 19.134.68.69 (k145n30) resigned as manager for fs1.
GPFS: 6027-630 [N] Node 19.134.68.70 (k145n32) appointed as manager for fs1.
```

For more information, see **`mmchmgr` command** in *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

[Displaying cluster configuration information](#)

Use the **`mmIscluster`** command to display cluster configuration information.

[Security mode](#)

The security mode of a cluster determines the level of security that the cluster provides for communications between nodes in the cluster and also for communications between clusters.

[Minimum release level of a cluster](#)

The minimum release level of a cluster is the currently enabled level of functionality of the cluster. It is expressed as an IBM Storage Scale version number, such as 5.1.7.

[Cluster quorum with quorum nodes](#)

Cluster quorum defines the minimum number of GPFS quorum nodes that must have the GPFS daemon actively running on them for the cluster to be operational. This number is half of the defined GPFS quorum nodes plus one, so it is a good idea to have an odd number of GPFS quorum nodes defined. Cluster quorum using only GPFS quorum nodes is the default quorum algorithm.

[Cluster quorum with quorum nodes and tiebreaker disks](#)

To use cluster quorum with quorum nodes and tiebreaker disks, more requirements apply.

[Configuring cluster configuration repository](#)

The cluster configuration repository (CCR) is an internal configuration store for various IBM Storage Scale components. It is not meant to be used directly by the customer.

Related tasks

[Creating your GPFS cluster](#)

You must first create a GPFS cluster by issuing the **`mmcrccluster`** command.

[Adding nodes to a GPFS cluster](#)

You can add nodes to an existing GPFS cluster by issuing the **`mmaddnode`** command or by using the IBM Storage Scale installation toolkit. The new nodes are available immediately after the successful completion of the process.

[Deleting nodes from a GPFS cluster](#)

You can delete nodes from a GPFS cluster by issuing the **`mmdeinode`** command.

[Changing the GPFS cluster configuration data](#)

You can use the **`mmchcluster`** or **`mmchconfig`** command to change the configuration attributes.

[Running IBM Storage Scale commands without remote root login](#)

With sudo wrapper scripts you can avoid configuring nodes to allow remote root login.

[Starting and stopping GPFS](#)

You can use the **`mmstartup`** and **`mmshutdown`** commands to start and stop GPFS on new or existing clusters.

[Shutting down an IBM Storage Scale cluster](#)

You might need to shut down an IBM Storage Scale cluster in an emergency.

Starting and stopping GPFS

You can use the **`mmstartup`** and **`mmshutdown`** commands to start and stop GPFS on new or existing clusters.

For new GPFS clusters, see *Steps to establishing and starting your GPFS cluster* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

For existing GPFS clusters, before starting GPFS, ensure that you have:

1. Verified the installation of all prerequisite software.
2. Compiled the GPL layer, if Linux is being used.

Tip: You can configure a cluster to rebuild the GPL automatically whenever a new level of the Linux kernel is installed or whenever a new level of IBM Storage Scale is installed. This feature is available only on the Linux operating system. For more information, see the description of the **autoBuildGPL** attribute in the topic *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

3. Properly configured and tuned your system for use by GPFS. This should be done prior to starting GPFS.

For more information, see [Chapter 4, “Configuring and tuning your system for GPFS,” on page 65](#).

Start the daemons on all of the nodes in the cluster by issuing the **mmstartup -a** command:

```
mmstartup -a
```

The output is similar to this:

```
Thu Nov 26 06:35:49 MST 2020: mmstartup: Starting GPFS ...
```

Check the messages recorded in `/var/adm/ras/mmfs.log.latest` on one node for verification. Look for messages similar to this:

```
2020-11-26_06:36:13.534-0700: [N] mmfsd ready
```

This indicates that quorum has been formed and this node has successfully joined the cluster, and is now ready to mount file systems.

If GPFS does not start, see *GPFS daemon will not come up in IBM Storage Scale: Problem Determination Guide*.

For more information, see **mmstartup command** in *IBM Storage Scale: Command and Programming Reference Guide*.

If it becomes necessary to stop GPFS, you can do so from the command line by issuing the **mmshutdown** command:

```
mmshutdown -a
```

The system displays information similar to:

```
Thu Nov 26 06:32:43 MST 2020: mmshutdown: Starting force unmount of GPFS file systems
Thu Nov 26 06:32:48 MST 2020: mmshutdown: Shutting down GPFS daemons
Thu Nov 26 06:32:59 MST 2020: mmshutdown: Finished
```

For more information, see **mmshutdown command** in *IBM Storage Scale: Command and Programming Reference Guide*.

Note: Before you shut down the current cluster manager node, run the **mmchmgr** command to move the cluster manager role to another node to avoid unexpected I/O interruption.

Related concepts

[Displaying cluster configuration information](#)

Use the **mmIscluster** command to display cluster configuration information.

[Security mode](#)

The security mode of a cluster determines the level of security that the cluster provides for communications between nodes in the cluster and also for communications between clusters.

[Minimum release level of a cluster](#)

The minimum release level of a cluster is the currently enabled level of functionality of the cluster. It is expressed as an IBM Storage Scale version number, such as 5.1.7.

[Cluster quorum with quorum nodes](#)

Cluster quorum defines the minimum number of GPFS quorum nodes that must have the GPFS daemon actively running on them for the cluster to be operational. This number is half of the defined GPFS quorum

nodes plus one, so it is a good idea to have an odd number of GPFS quorum nodes defined. Cluster quorum using only GPFS quorum nodes is the default quorum algorithm.

Cluster quorum with quorum nodes and tiebreaker disks

To use cluster quorum with quorum nodes and tiebreaker disks, more requirements apply.

Configuring cluster configuration repository

The cluster configuration repository (CCR) is an internal configuration store for various IBM Storage Scale components. It is not meant to be used directly by the customer.

Related tasks

Creating your GPFS cluster

You must first create a GPFS cluster by issuing the **mmcrcluster** command.

Adding nodes to a GPFS cluster

You can add nodes to an existing GPFS cluster by issuing the **mmaddnode** command or by using the IBM Storage Scale installation toolkit. The new nodes are available immediately after the successful completion of the process.

Deleting nodes from a GPFS cluster

You can delete nodes from a GPFS cluster by issuing the **mmdeinode** command.

Changing the GPFS cluster configuration data

You can use the **mmchcluster** or **mmchconfig** command to change the configuration attributes.

Running IBM Storage Scale commands without remote root login

With sudo wrapper scripts you can avoid configuring nodes to allow remote root login.

Displaying and changing the file system manager node

In general, GPFS performs the same functions on all nodes. There are also cases where one node provides a more global function that affects the operation of multiple nodes. For example, each file system is assigned a node that functions as a file system manager.

Shutting down an IBM Storage Scale cluster

You might need to shut down an IBM Storage Scale cluster in an emergency.

Starting or stopping GPFS daemon on a node by using GUI

You can start and shut down the GPFS services on the nodes. You must be careful with shutting down the GPFS services on a node because it may not only affect the functioning of the node but also the entire cluster depending on the special roles of the nodes in the cluster.

Note: When shutting down the GPFS service on a node, consider the following consequences:

- Unmounts all file systems and disrupt services on the node.
- If the node is a quorum node, then the cluster may lose quorum. To maintain quorum, you need to be certain that the number of available quorum nodes is equal to the total number of quorum nodes divided by two plus one (total number of quorum nodes/2 + 1). Or, you need to properly configure tiebreaker disks.

Perform the following steps to start and shut down GPFS daemon through IBM Storage Scale GUI:

1. Go to **Services > GPFS Daemon** page in the IBM Storage Scale GUI.
2. To start GPFS daemon on a node, select the node on which you need to start the GPFS daemon and click **Start Up**.
3. To stop GPFS daemon on a node, select the node on which you need to shut down GPFS daemon and click **Shut Down**.

Shutting down an IBM Storage Scale cluster

You might need to shut down an IBM Storage Scale cluster in an emergency.

1. Stop the protocol services on all protocol nodes in the cluster by using the **mmces service stop** command.

For example:

```
mmces service stop nfs -a  
mmces service stop smb -a  
mmces service stop s3 -a
```

2. Unmount all file systems, except the CES shared root file system, on all nodes in the cluster by using the **mmumount** command.
3. Stop GPFS daemons on all protocol nodes in the cluster by using the **mmshutdown -N cesNodes** command.
4. Unmount all file systems on all nodes in the cluster by using the **mmumount all -a** command.
5. Stop GPFS daemons on all nodes in the cluster by using the **mmshutdown -a** command.

After performing these steps, depending on your operating system, shut down your servers.

Before shutting down and powering up your servers, consider the following points:

- You must shut down NSD servers before the storage subsystem. While powering up, the storage subsystem must be online before NSD servers are up so that LUNs are visible to them.
- In a power-on scenario, verify that all network and storage subsystems are fully operational before bringing up any IBM Storage Scale nodes.
- On the Power® platform, you must shut down operating systems for LPARs first and then power off servers by using Hardware Management Console (HMC). HMC must be the last to be shut down and the first to be powered up.
- It is preferable to shut down your Ethernet and InfiniBand switches using the management console instead of powering them off. In any case, network infrastructure such as switches or extenders must be powered off last.
- After starting up again, verify that functions, such as AFM and policies are operational. You might need to manually restart some functions.
- There are a number other GPFS functions that could be interrupted by a shutdown. Ensure that you understand what else might need to be verified depending on your environment.

Related concepts

[Displaying cluster configuration information](#)

Use the **mmIscluster** command to display cluster configuration information.

Security mode

The security mode of a cluster determines the level of security that the cluster provides for communications between nodes in the cluster and also for communications between clusters.

Minimum release level of a cluster

The minimum release level of a cluster is the currently enabled level of functionality of the cluster. It is expressed as an IBM Storage Scale version number, such as 5.1.7.

Cluster quorum with quorum nodes

Cluster quorum defines the minimum number of GPFS quorum nodes that must have the GPFS daemon actively running on them for the cluster to be operational. This number is half of the defined GPFS quorum nodes plus one, so it is a good idea to have an odd number of GPFS quorum nodes defined. Cluster quorum using only GPFS quorum nodes is the default quorum algorithm.

Cluster quorum with quorum nodes and tiebreaker disks

To use cluster quorum with quorum nodes and tiebreaker disks, more requirements apply.

Configuring cluster configuration repository

The cluster configuration repository (CCR) is an internal configuration store for various IBM Storage Scale components. It is not meant to be used directly by the customer.

Related tasks

[Creating your GPFS cluster](#)

You must first create a GPFS cluster by issuing the **mmcrccluster** command.

[Adding nodes to a GPFS cluster](#)

You can add nodes to an existing GPFS cluster by issuing the **mmaddnode** command or by using the IBM Storage Scale installation toolkit. The new nodes are available immediately after the successful completion of the process.

Deleting nodes from a GPFS cluster

You can delete nodes from a GPFS cluster by issuing the **mmde1node** command.

Changing the GPFS cluster configuration data

You can use the **mmchcluster** or **mmchconfig** command to change the configuration attributes.

Running IBM Storage Scale commands without remote root login

With sudo wrapper scripts you can avoid configuring nodes to allow remote root login.

Displaying and changing the file system manager node

In general, GPFS performs the same functions on all nodes. There are also cases where one node provides a more global function that affects the operation of multiple nodes. For example, each file system is assigned a node that functions as a file system manager.

Starting and stopping GPFS

You can use the **mmstartup** and **mmshutdown** commands to start and stop GPFS on new or existing clusters.

Configuring cluster configuration repository

The cluster configuration repository (CCR) is an internal configuration store for various IBM Storage Scale components. It is not meant to be used directly by the customer.

For more information about the functionality of CCR, see *Clustered configuration repository in IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Related concepts

Displaying cluster configuration information

Use the **mm1scluster** command to display cluster configuration information.

Security mode

The security mode of a cluster determines the level of security that the cluster provides for communications between nodes in the cluster and also for communications between clusters.

Minimum release level of a cluster

The minimum release level of a cluster is the currently enabled level of functionality of the cluster. It is expressed as an IBM Storage Scale version number, such as 5.1.7.

Cluster quorum with quorum nodes

Cluster quorum defines the minimum number of GPFS quorum nodes that must have the GPFS daemon actively running on them for the cluster to be operational. This number is half of the defined GPFS quorum nodes plus one, so it is a good idea to have an odd number of GPFS quorum nodes defined. Cluster quorum using only GPFS quorum nodes is the default quorum algorithm.

Cluster quorum with quorum nodes and tiebreaker disks

To use cluster quorum with quorum nodes and tiebreaker disks, more requirements apply.

Related tasks

Creating your GPFS cluster

You must first create a GPFS cluster by issuing the **mmcrcluster** command.

Adding nodes to a GPFS cluster

You can add nodes to an existing GPFS cluster by issuing the **mmaddnode** command or by using the IBM Storage Scale installation toolkit. The new nodes are available immediately after the successful completion of the process.

Deleting nodes from a GPFS cluster

You can delete nodes from a GPFS cluster by issuing the **mmde1node** command.

Changing the GPFS cluster configuration data

You can use the **mmchcluster** or **mmchconfig** command to change the configuration attributes.

Running IBM Storage Scale commands without remote root login

With sudo wrapper scripts you can avoid configuring nodes to allow remote root login.

Displaying and changing the file system manager node

In general, GPFS performs the same functions on all nodes. There are also cases where one node provides a more global function that affects the operation of multiple nodes. For example, each file system is assigned a node that functions as a file system manager.

Starting and stopping GPFS

You can use the **mmstartup** and **mmshutdown** commands to start and stop GPFS on new or existing clusters.

Shutting down an IBM Storage Scale cluster

You might need to shut down an IBM Storage Scale cluster in an emergency.

Enabling CCR

Cluster configuration repository (CCR) is the cluster configuration method used in the IBM Storage Scale cluster. It is enabled by default on the cluster. If CCR is not enabled on the cluster, you can run the **mmchcluster --ccr-enable** command to enable it.

Note: Cluster configuration using primary and secondary server is deprecated and has been removed.

In IBM Storage Scale, unless CCR is enabled on the cluster, you cannot run the **mmchconfig release=LATEST** command to change the minimum release level to the latest version.

CCR directory structure and recommendations for configuring CCR

The working directory of CCR is at `/var/mmfs/CCR` on every node in the cluster. Directory structure on quorum node and non-quorum node is slightly different.

On quorum nodes, the following directory structure is available inside this directory:

```
# ls -al /var/mmfs/CCR
total 32
drwxr-xr-x. 4 root root 4096 Jul  8 16:00 .
drwxr-xr-x. 9 root root 4096 Jul  8 16:00 ..
drwxr-xr-x. 2 root root 4096 Jul  8 16:00 cached
-rw-r--r--. 1 root root    0 Jul  8 16:00 ccr.disks
-rw-r--r--. 1 root root    4 Jul  8 16:00 ccr.noauth
-rw-r--r--. 1 root root  114 Jul  8 16:00 ccr.nodes
-rw-----. 1 root root 4096 Jul  8 16:00 ccr.paxos.1
-rw-----. 1 root root 4096 Jul  8 16:00 ccr.paxos.2
drwxr-xr-x. 2 root root 4096 Jul  8 16:00 committed
```

On non-quorum node, the files like `ccr.paxos.1`, `ccr.paxos.2`, and `ccr.disks` do not exist.

The directory structure of all CCR files is also known as the CCR state.

The `cached` directory might contain a file that is named as `ccr.paxos`. This file is used by the CCR clients to speed up the process for locating quorum nodes. This file is a binary file.

The `ccr.disks` and `ccr.nodes` files are the configuration files of CCR. The `ccr.disks` file contains the CCR configuration of tiebreaker disks if the cluster is configured with tiebreaker disks. The `ccr.nodes` file contains the CCR configuration of the quorum nodes. Without having this file, the CCR client and server cannot establish a connection to exchange data with the CCR server on other quorum nodes. The master copies of both these files are stored in the `committed` directory on all quorum nodes as shown in the following example:

```
# ls -al /var/mmfs/CCR/committed/ccr.*
-rw-r--r--. 1 root root    0 Jul  8 16:00 /var/mmfs/CCR/committed/ccr.disks.2.2.ffffffff.01076f
-rw-r--r--. 1 root root 114 Jul  8 16:00 /var/mmfs/CCR/committed/ccr.nodes.1.7.f20ea9e3.01076e
```

The `ccr.paxos.1` and `ccr.paxos.2` files are used by the CCR to reach consensus among the quorum nodes, if an update happens to the CCR. At least one good copy of these files must be available. It can be used if the last write that modified these files failed for some reason. Both files are binary files.

The *committed* directory contains the files that are committed to the CCR. Changing those files manually might result in damaging CCR operation and state.

A complete copy of the CCR directory is available on every quorum node. The number of quorum nodes is limited to eight. Use an odd number of quorum nodes when no tiebreaker disks are configured.

CCR can be configured with up to three tiebreaker disks. It is strongly recommended to use an odd number of tiebreaker disks. The space that is reserved for the CCR is limited on these tiebreaker disks. Therefore, only the CCR Paxos state, which has nearly the same content of the `ccr.paxos.1` and `ccr.paxos.2` files are stored on the tiebreaker disks. That is, the files that are committed to the CCR are not stored on tiebreaker disks and it leads to a limitation for the combination of two quorum nodes and tiebreaker disks. For more information, see “[Limitations of CCR](#)” on page 41.

The CCR service remains available even if GPFS is shut down on a particular quorum node. This capability is achieved by including the CCR server functions in the following two GPFS daemon executable:

- A CCR server runs inside the `mmfsd` daemon when GPFS is up.
- A CCR server instance runs inside the `mmsdrserv` daemon when GPFS is down.

One of those daemons is running all the time on at least the quorum nodes. The only exceptions to this are the cases in which the `mmsdrserv` must be stopped by design. For example, in disaster-recovery scenarios.

Nearly all IBM Storage Scale administrator commands use the CCR to evaluate whether a new GPFS configuration file is available. This means that when the CCR is not working, most of the IBM Storage Scale administrator commands also do not work and the CCR must return into a working state first to run any administrator commands.

CCR uses the GSKit security library for authentication purposes.

Disaster recovery scenarios for CCR

Different procedures can be followed for recovering from a broken CCR. The recovery actions to be applied vary based on the use cases.

The following list provides the use cases:

Recovering from a single quorum or non-quorum node failure

A node failure might occur when the node is completely corrupted or when the node was rebuilt from scratch. In this scenario, just one quorum node is broken but there are still enough quorum nodes available on which CCR is running without any issue. This case must be even applied when a single non-quorum node must be recovered.

Command to apply to restore the configuration information: `mmsdrrestore -p <GOOD_QUORUM_NODE>`.

Recovering from the loss of a majority of quorum nodes

In this case, a majority of quorum nodes are broken but there is still at least one quorum node available with an intact CCR state.

Command to apply to restore the configuration information: `mmchnode --noquorum -N <LIST_OF_BROKEN_QUORUM_NODES> --force`

Recovering from damage or loss of the CCR on all quorum nodes

In this case, the CCR is partially broken on all quorum nodes. This means that there are still fragments of the CCR state available on various quorum nodes but no quorum node is available with a complete, intact CCR state.

Command to apply to restore the configuration information: `mmsdrrestore --ccr-repair`

Recovering from an existing CCR backup

In this case, the CCR state on all quorum nodes is lost. That is, loss of access to `/var/mmfs` or `/var/mmfs/ccr` on all quorum nodes. This assumes that a valid CCR backup is still available.

Command to apply to restore the configuration information: **mmsdrrestore -F <PATH_TO_CCR_BACKUP_FILE> -a**

The following sections describe the different recovery cases in more detail with examples.

Limitations of CCR

Ensure that you are aware of the following limitations of the Cluster configuration repository (CCR) to identify the workarounds, if any.

CCR limitation when the cluster is configured with two quorum nodes and at least one tiebreaker disk

You might need to shut down quorum nodes during a maintenance process. In a cluster with two quorum nodes, the cluster might not be able to reach quorum even when if one quorum node is active and it has access to the tiebreaker disks.

The reason for this limitation is that the CCR stores the committed files and the file updates only on quorum nodes and not on tiebreaker disks. After one quorum node becomes active, the CCR server on that quorum node reads the Paxos state from the tiebreaker disks during startup. This process might find out an occurrence of file update in the past. This file update went only to the other quorum node, which is not available when this quorum node is started up. This action results in no CCR quorum during startup.

Recommendation to avoid this limitation

Shut down quorum nodes one at a time. That is, shut down the second quorum node only after the first quorum node is started up and GPFS is in active state on the first quorum node. Use the **mmchmgr** command to assign the cluster manager role to the quorum node that remains active before you shut down the other quorum node.

The following example shows the recommended procedure. The cluster that is used in this example is configured with two quorum nodes and one tiebreaker disk:

```
# mm1scluster

GPFS cluster information
=====
GPFS cluster name: gpfs-cluster-2.localnet.com
GPFS cluster id: 13445038716777666550
GPFS UID domain: gpfs-cluster-2.localnet.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 node-21.localnet.com 10.0.100.21 node-21.localnet.com quorum
2 node-22.localnet.com 10.0.100.22 node-22.localnet.com quorum
3 node-23.localnet.com 10.0.100.23 node-23.localnet.com
4 node-24.localnet.com 10.0.100.24 node-24.localnet.com
5 node-25.localnet.com 10.0.100.25 node-25.localnet.com

# mm1sconfig tiebreakerDisks
tiebreakerDisks disk1
```

All nodes in the cluster are active and the current cluster manager is *node-21*:

```
# mmgetstate -a

Node number Node name GPFS state
-----
1 node-21 active
2 node-22 active
3 node-23 active
4 node-24 active
5 node-25 active

# mm1smgr
file system manager node
```

```
-----  
gpfs0          10.0.100.21 (node-21)  
Cluster manager node: 10.0.100.21 (node-21)
```

The following example shows that the second quorum node *node-22* is shut down for maintenance purposes and the remaining quorum node remains active:

```
# mmgetstate -a  
  
Node number  Node name  GPFS state  
-----  
    1  node-21    active  
    2  node-22    unknown  
    3  node-23    active  
    4  node-24    active  
    5  node-25    active
```

After the maintenance for quorum node *node-22* is completed, it rejoins the cluster and becomes active again:

```
# mmgetstate -a  
  
Node number  Node name  GPFS state  
-----  
    1  node-21    active  
    2  node-22    active  
    3  node-23    active  
    4  node-24    active  
    5  node-25    active
```

To shut down quorum node *node-21* for maintenance purposes with minimal disruption, assign the quorum node *node-22* as the new cluster manager. If node *node-21* is the file system manager for GPFS file systems, you also need to assign it as the new file system manager:

```
# mmchmgr -c node-22  
Appointing node 10.0.100.22 (node-22) as cluster manager  
Node 10.0.100.22 (node-22) has taken over as cluster manager
```

```
# mmchmgr gpfs0 node-22  
Sending migrate request to current manager node 10.0.100.21 (node-21).  
Node 10.0.100.21 (node-21) resigned as manager for gpfs0.  
Node 10.0.100.22 (node-22) appointed as manager for gpfs0.
```

```
# mmlsmgr  
file system      manager node  
-----  
gpfs0            10.0.100.22 (node-22)  
  
Cluster manager node: 10.0.100.22 (node-22)
```

Now quorum node *node-21* can be shut down for maintenance purposes without losing the GPFS quorum:

```
# mmgetstate -a  
  
Node number  Node name  GPFS state  
-----  
    1  node-21    unknown  
    2  node-22    active  
    3  node-23    active  
    4  node-24    active  
    5  node-25    active
```

After maintenance is done for quorum node *node-21*, it rejoins the cluster and becomes active again as shown in the following example:

```
# mmgetstate -a  
  
Node number  Node name  GPFS state  
-----  
    1  node-21    active  
    2  node-22    active
```

```
3 node-23 active
4 node-24 active
5 node-25 active
```

CCR Limitation on using Persistent Reserve (PR) when a disk is already set as a CCR tiebreaker disk

On AIX and Linux, if the CCR is configured with tiebreaker disks, the **mmchconfig** command fails when setting the *usePersistentReserve* flag, as shown in the following example:

```
# mmchconfig usePersistentReserve=yes
Verifying GPFS is stopped on all nodes ...
mmchconfig: Processing disk gpfs1nsd
mmchconfig: chdev failed to set PR_key_value for hdisk2.
mmchconfig: 6027-1940 Unable to set reserve policy PR_shared on disk gpfs1nsd on node
node-22.localnet.com.
mmchconfig: 6027-1214 Unable to enable Persistent Reserve on the following disks:
gpfs1nsd
mmchconfig: The usePersistentReserve parameter will remain unchanged.
mmchconfig: 6027-1639 Command failed. Examine previous error messages to determine cause.
```

CCR uses the tiebreaker disks and opens access to those disks on the quorum nodes to use them for the Paxos protocol. This limitation prevents the **mmchconfig** command from setting the Persistent Reserve (PR) flag successfully.

The workaround to set the PR flag successfully is as follows:

- Configure the CCR without tiebreaker disks, by using the **mmchconfig tiebreakerDisks=no** command.
- Set the PR flag, by using the **mmchconfig usePersistentReserve=yes**.
- Reconfigure the CCR with the original tiebreaker disks, by using the **mmchconfig tiebreakerDisks=<ORIGINAL_TIEBREAKER_DISKS>**.

Chapter 2. Configuring GPUDirect Storage for IBM Storage Scale

After IBM Storage Scale is installed, the GPUDirect Storage (GDS) feature can be enabled by running the command **mmchconfig verbsGPUDirectStorage=yes**. This requires that IBM Storage Scale is stopped on all nodes.

You also need to set the following configuration options by using the **mmchconfig** command on the GDS clients and storage servers:

- minReleaseLevel must be 5.1.2 or later.
- verbsRdma=enable
- verbsRdmaSend=yes
- verbsPorts. The values must be compliant with the values of the rdma_dev_addr_list parameter that is at /etc/cufile.json, i.e. the IP addresses assigned to rdma_dev_addr_list in /etc/cufile.json need to be assigned to the RDMA devices listed in the IBM Storage Scale config variable verbsPorts. For more information, see the “[Configuring RDMA ports on the GPU client](#)” on page 46 section below.

Configuring virtual fabrics

The RDMA subsystem within IBM Storage Scale is supporting virtual fabrics to control how RDMA ports on NSD clients and NSD servers communicate with each other through Queue Pairs. Only RDMA ports on the same virtual fabric communicate with each other. With this feature, it is possible to use GDS on setups with multiple separated InfiniBand fabrics.

An example of a virtual fabric definition with the virtual fabrics 0 and 1 is shown as follows:

```
verbsPorts mlx5_4/1/0 mlx5_5/1/0 mlx5_10/1/1 mlx5_11/1/1
```

For more information on *Virtual fabrics*, see the topic *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Virtual fabric numbers used on GDS clients must be used on **all** NSD servers. Otherwise an error is thrown if an NSD server cannot reach the GPU client within its virtual fabric. There are no special configuration changes within the NVIDIA GDS software stack required for virtual fabrics. All IP addresses configured in /etc/cufile.json for the key rdma_dev_addr_list must be reachable by the NSD servers. The verbsPorts configuration variable needs to be set accordingly. If GDS I/O operations go through an RDMA port not listed in verbsPorts, it results in an I/O error and an error message is logged in the IBM Storage Scale log file. The verbsPorts syntax remains unchanged.

All NSD servers must have RDMA ports in all virtual fabrics that are configured on the NSD clients that perform I/O through GDS. For example, on the GDS clients, the RDMA ports are configured to use virtual fabric numbers 1, 2, 3, and 4. On the NSD server, RDMA ports with the same 4 virtual fabric numbers must be configured. When a GDS client submits a GDS request through an RDMA port on the virtual fabric number 4, but the NSD server does not have an RDMA port on virtual fabric number 4, the request fails and results in an I/O error in the GDS application. An error message in the IBM Storage Scale log file also gets recorded.

Configuring CUDA

The configuration file (“/etc/cufile.json”) for CUDA and the GDS driver can be found on each GDS client.

Note: This topic describes the configuration that is necessary for IBM Storage Scale. For an in-depth discussion of these configuration options, see [Installing GDS](#).

You need to update the following configuration settings in the CUDA file:

rdma_dev_addr_list

Defines the RDMA devices to be used as a list of IP addresses. The IP addresses (RoCE or IP over IB) specified must be consistent with the values that are set for the verbsPorts parameter on the GDS clients. For more information, see the “[Configuring RDMA ports on the GPU client](#)” on page 46 section below.

rdma_load_balancing_policy

Specifies the load-balancing policy for RDMA memory registration. If the GDS client is a DGX, the following values must be set:

- RoundRobin: For storage Network Interface Cards (NIC).
- RoundRobinMaxMin: For compute NICs.

The default value is RoundRobin. For more information on DGX, see <https://www.nvidia.com/en-us/data-center/dgx-systems/>.

rdma_access_mask

Enables relaxed ordering. Set the value 0x1f.

"logging"."level"

Defines the log level. Set the values ERROR or WARN unless debug output is required. Setting log levels such as DEBUG and TRACE impacts performance.

use_poll_mode

Switches the NVIDIA driver between asynchronous and synchronous I/O modes. Set the value `false` for configuring GDS for IBM Storage Scale.

gds_write_support

For accelerated writes the following key/value has to be added in the file system-specific section ("fs": "gpfs"):

```
"gpfs": {  
    "gds_write_support": true  
}
```

Configuring RoCE

To enable RoCE for GDS, enable the general RoCE support for IBM Storage Scale. No special configuration settings are needed to enable the RoCE support for GDS.

To enable generic RoCE support, all RoCE adapter ports must have a proper IP configuration and these ports must be listed in the verbsPorts configuration variable. In addition, the verbsRdmaCm configuration variable must be enabled. This setting enables the RDMA Connection Manager, which is a prerequisite for using RoCE.

For more information, see [Highly Efficient Data Access with RoCE on IBM Elastic Storage® Systems and IBM Storage Scale](#).

To configure the CUDA software stack, the configuration file `/etc/cufile.json` must have in the key `rdma_dev_addr_list` all or some of the IP addresses for the RoCE ports configured in the GPFS `verbsPorts` configuration variable.

Configuring RDMA ports on the GPU client

Configuring the RDMA ports requires the following steps:

1. Configuration of the RDMA ports to be used by IBM Storage Scale and GDS on the GPU client machine.

Specify the RDMA ports to be used in the `verbsPorts` config option, for example:

```
root:~# mmIsconfig verbsports  
verbsPorts mlx5_4/1 mlx5_5/1 mlx5_10/1 mlx5_11/1
```

These are the ports used by IBM Storage Scale and they can also be used by GDS. GDS can use all ports but does not have to.

2. Determine the IP addresses for the RDMA ports.

Determine the device names by using the **ibdev2netdev** command:

```
# ibdev2netdev
m1x5_4 port 1 ==> enp97s0f0 (Up)
m1x5_5 port 1 ==> enp97s0f1 (Up)
m1x5_10 port 1 ==> enp225s0f0 (Up)
m1x5_11 port 1 ==> enp225s0f1 (Up)
```

List the IP addresses assigned by using the **ip** command:

```
# ip -br -4 a
enp97s0f0      UP            192.168.1.20/24
enp97s0f1      UP            192.168.1.21/24
enp225s0f0      UP            192.168.1.22/24
enp225s0f1      UP            192.168.1.23/24
```

3. Use these IP addresses in the config /etc/cufile.json file:

The config file /etc/cufile.json has an entry for the RDMA device address list called `rdma_dev_addr_list`.

This should be set to all or some of the IPs found in the previous step.

```
"rdma_dev_addr_list": ["192.168.1.20", "192.168.1.21", "192.168.1.22", "192.168.1.23"],
```


Chapter 3. Configuring the CES and protocol configuration

After GPFS is configured, Cluster Export Services (CES) and its protocols can be configured, administered, or removed from the system.

Some of the CES and protocol configuration steps are already completed through the IBM Storage Scale installer. To verify, see the information about the IBM Storage Scale installer and protocol configuration in the topic *spectrumscale command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

A manual or a minimal installation of CES involves configuration and administrative tasks.

Configuring Cluster Export Services

Configure Cluster Export Services (CES) if it is not already configured through the installer.

For more information, see [Chapter 45, “Implementing Cluster Export Services,” on page 651](#).

Setting up Cluster Export Services shared root file system

If a shared root file system through the installer is not set up, you must create one for Cluster Export Services (CES).

The CES shared root (`cesSharedRoot`) is needed for storing CES shared configuration data, for protocol recovery, and for other protocol-specific purposes. It is part of the cluster export configuration and is shared between the protocols. Every CES node requires access to the path configured as shared root.

The `mmchconfig` command is used to configure this directory as part of setting up a CES cluster.

The `cesSharedRoot` cannot be changed while any of the CES nodes are up and running. Follow these steps to modify the shared root configuration:

1. Run the following command to suspend all CES nodes:

```
mmces suspend --stop -a
```

2. Run the following command to shut down all the CES nodes:

```
mmshutdown -a
```

3. Create `cesSharedRoot` by using the following command:

```
mmchconfig cesSharedRoot=/gpfs/fs0
```

4. Run the following command to start GPFS on all the CES nodes:

```
mmstartup -a
```

5. Run the following command to activate CES:

```
mmces resume --start -a
```

The `cesSharedRoot` is monitored by the system health daemon. If the shared root is not available, the CES node list command, `mmces node list` displays no-shared-root, and a failover is triggered.

The `cesSharedRoot` cannot be unmounted when the CES cluster is up and running. You need to bring all CES nodes down if you want to unmount `cesSharedRoot` (for example, for doing service action like `fsck`).

To list the current `cesSharedRoot`, run:

```
mmlsconfig cesSharedRoot
```

```
cesSharedRoot /gpfs/gpfs-ces/
```

The recommendation for CES shared root is a dedicated file system, but it is not enforced. It can also be a part (path) of an existing GPFS file system. A dedicated file system can be created with the **mmcrfs** command. In any case, CES shared root must reside on GPFS and must be available when it is configured through the **mmchconfig** command.

If not already done through the installer, it is recommended that you create a file system for the CES. Some protocol services share information through a cluster-wide file system. It is recommended to use a separate file system for this purpose.

Note: The recommended size for CES shared root file system is greater than or equal to 4 GB. It is recommended to use the following settings for the CES shared root file system if it is used for CES shared root only:

- Block size: 256 KB
- Starting inode-limit: 5000

To set up CES, change the configuration to use the new file system:

```
mmchconfig cesSharedRoot=/gpfs/fs0
```

Note:

- When the GPFS starts back up, as the cesSharedRoot is now defined, the CES can be enabled on the cluster.
- If file audit logging is already enabled for the file system that you defined for cesSharedRoot, you need to first disable and then enable it again for that file system.

```
mmaudit Device disable
```

```
mmaudit Device enable
```

Related concepts

[Configuring Cluster Export Services nodes](#)

If you do not configure Cluster Export Services (CES) nodes through the installer, you must configure them before you configure any protocols.

[Configuring CES protocol service IP addresses](#)

Protocol services are made available through Cluster Export Services (CES) protocol service IP addresses. These addresses are separate from the IP addresses that are used internally by the cluster.

[CES IP aliasing to network adapters on protocol nodes](#)

Cluster Export Services (CES) is a functionality in IBM Storage Scale that enables NFS and SMB protocols. Irrespective of which protocols you choose, all are accessible through a floating pool of IP addresses called CES IP addresses. This pool of CES IP addresses is considered floating because each IP can move independently among all protocol nodes. During a protocol node failure, accessibility to all protocols is maintained as the CES IP addresses automatically move from the failed protocol node to a healthy protocol node. Use this information to understand how CES IP addresses are assigned and are aliased to adapters with or without VLAN tagging.

[Deploying Cluster Export Services packages on existing IBM Storage Scale nodes](#)

Use the following instructions to copy packages on your protocol nodes and to deploy these packages.

[Verifying the final CES configurations](#)

After you finish configuring the Cluster Export Services (CES), you must verify the final configuration.

Configuring Cluster Export Services nodes

If you do not configure Cluster Export Services (CES) nodes through the installer, you must configure them before you configure any protocols.

If not already done during the installation, configuration of CES nodes must be done before you configure any protocols. Nodes that participate in the handling of protocol exports must be configured as CES nodes.

Note: A CES cluster has a maximum of 32 protocol nodes if only NFS is enabled and a maximum of 16 protocol nodes if both SMB and NFS are enabled.

For each of the nodes that handle protocol exports, run:

```
mmchnode -N nodename --ces-enable
```

After you configure all nodes, verify that the list of CES nodes is complete:

```
mmces node list
```

CES nodes can be assigned to CES groups. A CES group is identified by a group name that has lowercase alphanumeric characters. CES groups can be used to manage CES node and address assignments.

Nodes can be assigned to groups by issuing the following command:

```
mmchnode --ces-group group1 -N node
```

A node can be assigned to multiple groups by issuing the following command:

```
mmchnode --ces-group group1,group2,group3 -N node1,node2
```

The group assignment can also be specified when the node is enabled for CES by issuing the following command:

```
mmchnode --ces-enable --ces-group group1,group2 -N node
```

The node can be removed from a group at any time by issuing the following command:

```
mmchnode --no ces-group group1 -N node
```

For more information, see *mmchnode command* in *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

[Setting up Cluster Export Services shared root file system](#)

If a shared root file system through the installer is not set up, you must create one for Cluster Export Services (CES).

[Configuring CES protocol service IP addresses](#)

Protocol services are made available through Cluster Export Services (CES) protocol service IP addresses. These addresses are separate from the IP addresses that are used internally by the cluster.

[CES IP aliasing to network adapters on protocol nodes](#)

Cluster Export Services (CES) is a functionality in IBM Storage Scale that enables NFS and SMB protocols. Irrespective of which protocols you choose, all are accessible through a floating pool of IP addresses called CES IP addresses. This pool of CES IP addresses is considered floating because each IP can move independently among all protocol nodes. During a protocol node failure, accessibility to all protocols is maintained as the CES IP addresses automatically move from the failed protocol node to a healthy protocol node. Use this information to understand how CES IP addresses are assigned and are aliased to adapters with or without VLAN tagging.

[Deploying Cluster Export Services packages on existing IBM Storage Scale nodes](#)

Use the following instructions to copy packages on your protocol nodes and to deploy these packages.

Verifying the final CES configurations

After you finish configuring the Cluster Export Services (CES), you must verify the final configuration.

Configuring CES protocol service IP addresses

Protocol services are made available through Cluster Export Services (CES) protocol service IP addresses. These addresses are separate from the IP addresses that are used internally by the cluster.

Each CES protocol service IP address is assigned initially to one CES node, either explicitly as specified by the **mmces address add** command, or by the system. They can be moved later either manually or automatically in response to certain events. The sample command is shown:

```
mmces address add --ces-node Node1 --ces-ip 192.168.6.6
```

After you add the required CES protocol service IP addresses, you must verify the configuration:

```
mmces address list
```

Use **mmces address add --ces-ip 192.168.6.6** to add an IP address to the CES IP address pool. The IP address is assigned to a CES node according to the CES address distribution policy.

CES addresses can be assigned to CES groups. A CES group is identified by a group name that consists of alphanumeric characters, which are case-sensitive. You can assign addresses to a group when they are defined by issuing the following command:

```
mmces address add --ces-ip 192.168.6.6 --ces-group group1
```

You can change the group assignment by issuing the following command:

```
mmces address change --ces-ip 192.168.6.6 --ces-group group2
```

You can remove the group assignment by issuing the following command:

```
mmces address change --ces-ip 192.168.6.6 --remove-group
```

A CES address that is associated with a group must be assigned only to a node that is also associated with the same group. A node can belong to multiple groups while an address cannot.

As an example, consider a configuration with three nodes. All three nodes can host addresses on subnet A, and two of the nodes can host addresses on subnet B. The nodes must have an existing non-CES IP address of the same subnet that is configured on the interfaces that are intended to be used for the CES IPs. Also, four addresses are defined, two on each subnet.

```
Node1: groups=subnetA,subnetB  
Node2: groups=subnetA,subnetB  
Node3: groups=subnetA  
Address1: subnetA  
Address2: subnetA  
Address3: subnetB  
Address4: subnetB
```

In this example, Address1 and Address2 can be assigned to any of the three nodes, but Address3 and Address4 can be assigned to only Node1 or Node2.

If an address is assigned to a group for which there are no healthy nodes, the address remains unassigned until a node in the same group becomes available.

Addresses without a group assignment can be assigned to any node. Therefore, it is necessary to use a group for each subnet when multiple subnets exist.

Note: IP addresses that are assigned attributes (such as `object_database_node` or `object_singleton_node`) do not follow the same policy rules that other IP addresses follow. If a node has an affinity policy set, the IP address that is associated with the assigned attribute fails back to its node.

Configuring IPv6 addresses in the CES interface mode

If you want to use IPv6 addresses, you must use the CES interface mode. To use IPv6 addresses in your cluster, after the upgrade to IBM Storage Scale 5.1.x is completed, do the following steps:

1. Define the NICs by using the **mmces interface add** command.
2. Before switching to the interface mode, check the NIC configuration by using the **mmces interface check** command.
3. Switch to the interface mode by using the **mmces interface mode interface** command.
4. Add the IPv6 addresses in the Classless Inter-Domain Routing (CIDR) notation by using the **mmces address add** command.

Note: You can use the **mmces interface check** command at any time a problem is suspected to check the NIC configuration.

4. Add the IPv6 addresses in the Classless Inter-Domain Routing (CIDR) notation by using the **mmces address add** command.

Note: In the CES interface mode, you must add IP addresses only in the CIDR notation. For example:

```
IP:      1.2.3.4    or  123:124  
CIDR:   1.2.3.4/14 or  123::124/88
```

For information about the CIDR notation, see [Classless Inter-Domain Routing](#).

For more information, see *mmces command* in *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

[Setting up Cluster Export Services shared root file system](#)

If a shared root file system through the installer is not set up, you must create one for Cluster Export Services (CES).

[Configuring Cluster Export Services nodes](#)

If you do not configure Cluster Export Services (CES) nodes through the installer, you must configure them before you configure any protocols.

[CES IP aliasing to network adapters on protocol nodes](#)

Cluster Export Services (CES) is a functionality in IBM Storage Scale that enables NFS and SMB protocols. Irrespective of which protocols you choose, all are accessible through a floating pool of IP addresses called CES IP addresses. This pool of CES IP addresses is considered floating because each IP can move independently among all protocol nodes. During a protocol node failure, accessibility to all protocols is maintained as the CES IP addresses automatically move from the failed protocol node to a healthy protocol node. Use this information to understand how CES IP addresses are assigned and are aliased to adapters with or without VLAN tagging.

[Deploying Cluster Export Services packages on existing IBM Storage Scale nodes](#)

Use the following instructions to copy packages on your protocol nodes and to deploy these packages.

[Verifying the final CES configurations](#)

After you finish configuring the Cluster Export Services (CES), you must verify the final configuration.

CES IP aliasing to network adapters on protocol nodes

Cluster Export Services (CES) is a functionality in IBM Storage Scale that enables NFS and SMB protocols. Irrespective of which protocols you choose, all are accessible through a floating pool of IP addresses called CES IP addresses. This pool of CES IP addresses is considered floating because each IP can move independently among all protocol nodes. During a protocol node failure, accessibility to all protocols is maintained as the CES IP addresses automatically move from the failed protocol node to a healthy protocol node. Use this information to understand how CES IP addresses are assigned and are aliased to adapters with or without VLAN tagging.

Virtual LANs (VLANs) are often associated with secure networks because they provide a means of separating network devices into independent networks. Although the physical network infrastructure is

shared, unicast, multicast, and broadcast traffic from a network device in a VLAN is restricted to other devices within that same VLAN.

How are CES IP addresses assigned

CES IP addresses are automatically assigned and aliased to existing network adapters on protocol nodes during startup. The following example shows aliased CES IP addresses in a flat network environment or a single VLAN environment. The switch ports in these environments are set to Access mode and thus do not need VLAN tagging.

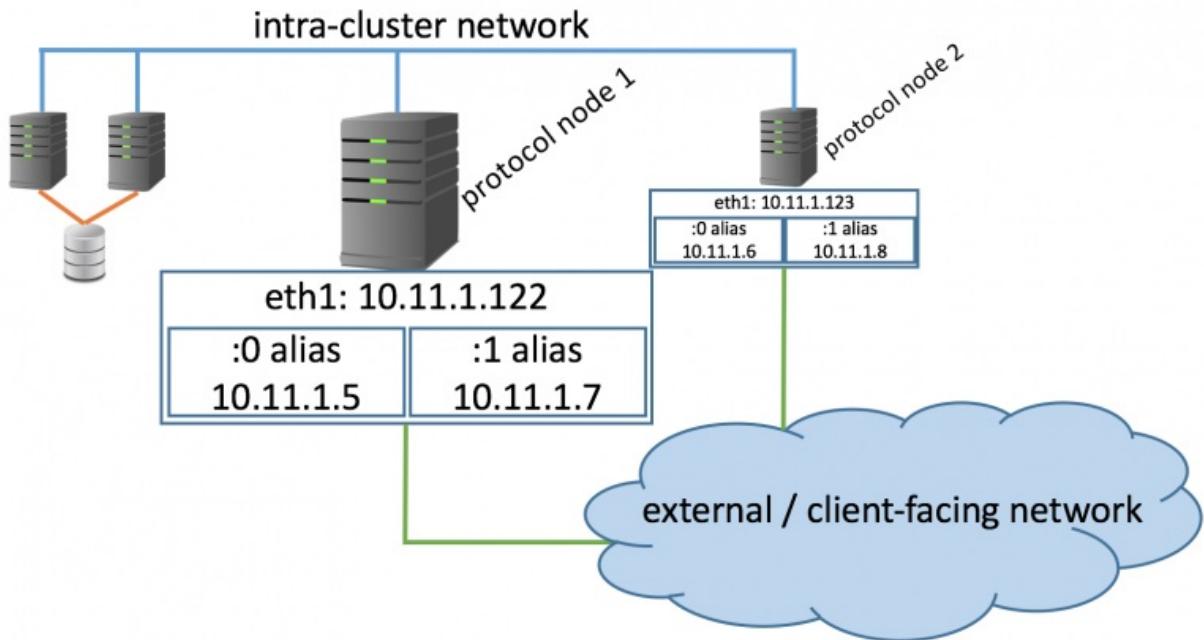


Figure 1. CES IPs in single VLAN environment

Example of aliased CES IP addresses by using the ip addr command

```
eth1: <POINTWISE,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq state UP qlen 1000
    link/ether 00:50:56:83:16:e5 brd ff:ff:ff:ff:ff:ff
    inet 10.11.1.122/24 brd 10.11.1.255 scope global eth1
        valid_lft forever preferred_lft forever
    inet 10.11.1.5/24 brd 10.11.1.255 scope global secondary eth1
        valid_lft forever preferred_lft forever
    inet 10.11.1.7/24 brd 10.11.1.255 scope global secondary eth1
        valid_lft forever preferred_lft forever
```

Example of preexisting routes

Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
default	gateway	0.0.0.0	UG	100	0	0	eth1
10.11.1.0	0.0.0.0	255.255.255.0	U	100	0	0	eth1
172.31.128.0	0.0.0.0	255.255.128.0	U	300	0	0	data0

In the preceding example, eth1 preexists with an established route and IP: 10.11.1.122. This IP is manually assigned and must be accessible before any CES configuration. When the CES services are active, CES IP addresses are then automatically aliased to this base adapter, thus creating eth1. The floating CES IP addresses assigned to the aliases are 10.11.1.5 and 10.11.1.7. Both CES IP addresses are allowed to move to other nodes if there is a failure. This automatic movement combined with the ability to manually move CES IP addresses, might cause a variance in the number of aliases and CES IP addresses among protocol nodes. The data0 interface illustrates how a network used for GPFS intra-cluster connectivity between nodes can be separate from the adapter that is used for CES IP addresses.

Example distribution of CES IP addresses among two protocol nodes after enablement of protocols

mmces address list			
Address	Node	Group	Attribute
10.11.1.5	protocol-node-1	none	none
10.11.1.6	protocol-node-2	none	object_database_node,object_singleton_node
10.11.1.7	protocol-node-1	none	none
10.11.1.8	protocol-node-2	none	none

CES IP addresses and VLAN tags

A network switch port can be considered a trunk port if it gives access to multiple VLANs. When it occurs, it is necessary for a VLAN tag to be added to each frame. This VLAN tag is an identification that allows switches to contain traffic within specific networks. If multiple networks must access data from IBM Storage Scale protocol nodes, then one possible option is to configure trunk ports on the switch that is directly connected to the IBM Storage Scale protocol nodes. After a trunk port is configured, VLAN tags are necessary on the connected network adapters. The CES IP addresses are automatically assigned and aliased to existing network adapters on protocol nodes during startup. To enable this process, the available VLAN tags require a preexisting network adapter with an established route and IP so that the CES IP addresses can alias to it.

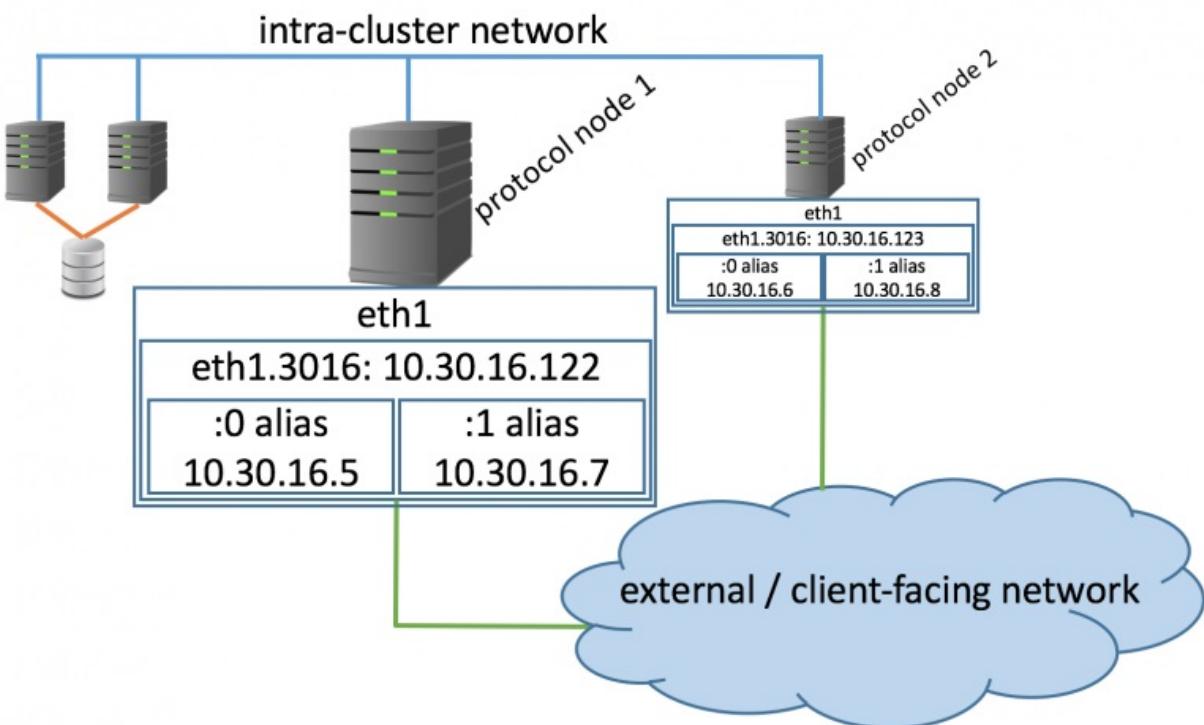


Figure 2. CES IPs and single VLAN tag

Example of aliased CES IP addresses by using the ip addr command (with VLAN tag)

```

eth1: <POINTOPOINT,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc noqueue state UNKNOWN
    link/ether 00:50:56:83:16:e5 brd ff:ff:ff:ff:ff:ff
    valid_lft forever preferred_lft forever

eth1.3016: <POINTOPOINT,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq state UP qlen 1000
    link/ether 00:50:56:83:16:e5 brd ff:ff:ff:ff:ff:ff
    inet 10.30.16.122/24 brd 10.30.16.255 scope global eth1.3016
        valid_lft forever preferred_lft forever
    inet 10.30.16.5/24 brd 10.30.16.255 scope global secondary eth1.3016
        valid_lft forever preferred_lft forever
    inet 10.30.16.7/24 brd 10.30.16.255 scope global secondary eth1.3016
        valid_lft forever preferred_lft forever

```

Example of pre-existing routes (with VLAN tag)

Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
default	gateway	0.0.0.0	UG	100	0	0	eth1.3016
10.30.16.0	0.0.0.0	255.255.255.0	U	100	0	0	eth1.3016
172.31.128.0	0.0.0.0	255.255.128.0	U	300	0	0	data0

As in the no VLAN tag example, an existing network adapter must be present so that CES\ IP addresses can alias to it. No IP addresses are assigned to the non-VLAN base adapter eth1. In this example, the preexisting network adapter with an established route and IP is eth1.3016. The IP for eth1.3016 is 10.30.16.122 and the VLAN tag is 3016. This preexisting IP can be used for network verification, before configuration of CES IP, by pinging it from external to the cluster or pinging it from other protocol nodes. It is a good practice to make sure that all protocol node base adapter IP addresses are accessible before the protocols are enabled. The data0 interface shows how a network used for GPFS intra-cluster connectivity between nodes can be separate from the adapter that is used for CES IP addresses.

Example distribution of CES IP addresses among two protocol nodes after enablement of protocols (with VLAN tag)

mmces address list			
Address	Node	Group	Attribute
10.30.16.5	protocol-node-1	none	none
10.30.16.6	protocol-node-2	none	
object_database_node,object_singleton_node			
10.30.16.7	protocol-node-1	none	none
10.30.16.8	protocol-node-2	none	none

CES IP addresses and multiple VLAN tags

The following diagram shows a node with two network adapters that are devoted to CES protocols: eth1 and eth2. Two VLANs are associated with the eth1 interface: 3016 and 3017. One VLAN is associated with the eth2 interface: 80.

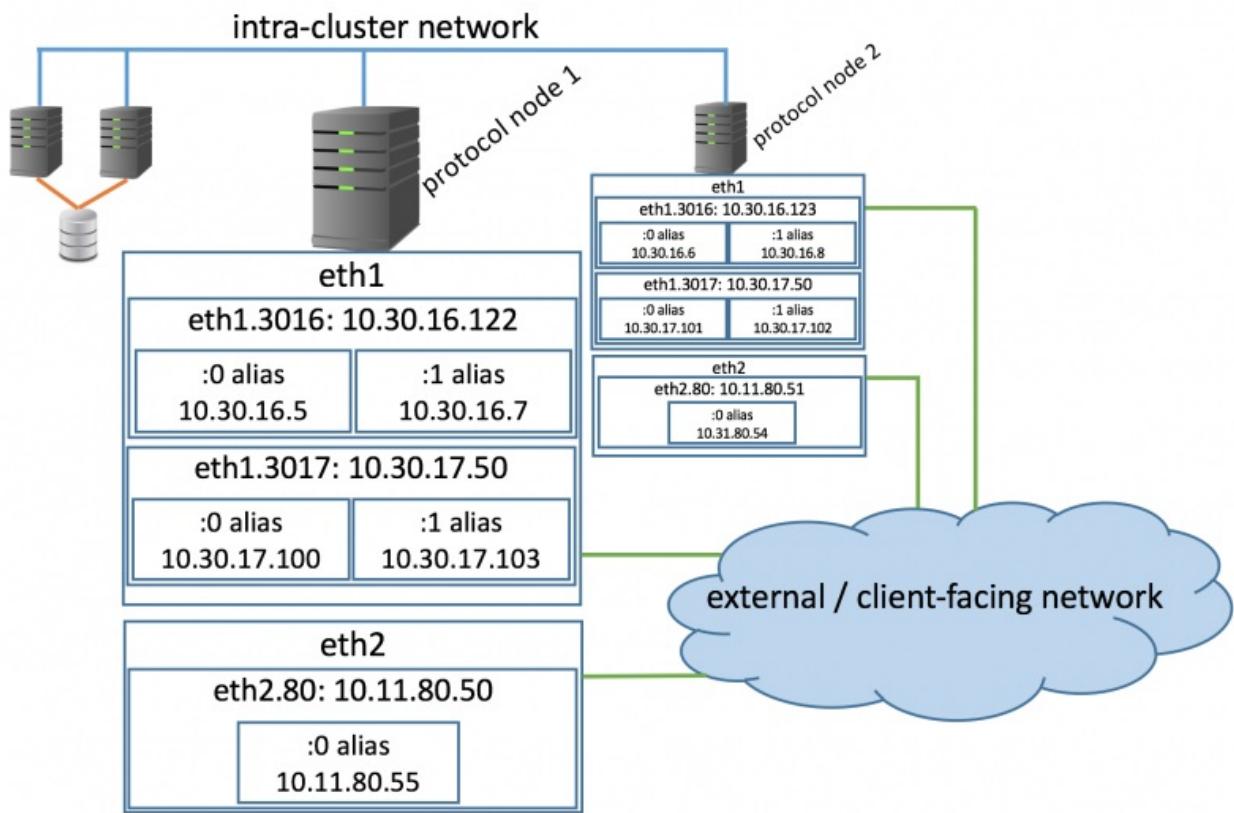


Figure 3. CES IPs and multiple VLAN tags

Example of aliased CES IP addresses by using the ip addr command (with multiple VLAN tags)

```

eth1: <POINTWISE,BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 9000 qdisc noqueue state UNKNOWN
    link/ether 00:50:56:83:16:e5 brd ff:ff:ff:ff:ff:ff
        valid_lft forever preferred_lft forever

eth1.3016: <POINTWISE,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq state UP qlen 1000
    link/ether 00:50:56:83:16:e5 brd ff:ff:ff:ff:ff:ff
        inet 10.30.16.122/24 brd 10.30.16.255 scope global eth1.3016
            valid_lft forever preferred_lft forever
        inet 10.30.16.5/24 brd 10.30.16.255 scope global secondary eth1.3016
            valid_lft forever preferred_lft forever
        inet 10.30.16.7/24 brd 10.30.16.255 scope global secondary eth1.3016
            valid_lft forever preferred_lft forever

eth1.3017: <POINTWISE,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq state UP qlen 1000
    link/ether 00:50:56:83:16:e5 brd ff:ff:ff:ff:ff:ff
        valid_lft forever preferred_lft forever
        inet 10.30.17.50/24 brd 10.30.17.255 scope global eth1.3017
            valid_lft forever preferred_lft forever
        inet 10.30.17.100/24 brd 10.30.17.255 scope global secondary eth1.3017
            valid_lft forever preferred_lft forever
        inet 10.30.17.103/24 brd 10.30.17.255 scope global secondary eth1.3017
            valid_lft forever preferred_lft forever

eth2: <POINTWISE,BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 9000 qdisc noqueue state UNKNOWN
    link/ether 00:50:56:83:16:e5 brd ff:ff:ff:ff:ff:ff
        valid_lft forever preferred_lft forever

eth2.80: <POINTWISE,BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq state UP qlen 1000
    link/ether 00:50:56:83:16:e5 brd ff:ff:ff:ff:ff:ff
        valid_lft forever preferred_lft forever
        inet 10.11.80.50/24 brd 10.11.80.255 scope global eth1.80
            valid_lft forever preferred_lft forever
        inet 10.11.80.55/24 brd 10.11.80.255 scope global secondary eth1.80
            valid_lft forever preferred_lft forever

```

Example of preexisting routes (with multiple VLAN tag)

Kernel IP routing table							
Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
default	gateway	0.0.0.0	UG	100	0	0	eth1.3016
10.30.16.0	0.0.0.0	255.255.255.0	U	100	0	0	eth1.3016
10.30.17.0	0.0.0.0	255.255.255.0	U	400	0	0	eth1.3017
10.11.80.0	0.0.0.0	255.255.255.0	U	200	0	0	eth2.80
172.31.128.0	0.0.0.0	255.255.128.0	U	300	0	0	data0

Example distribution of CES IP addresses from multiple VLANs among two protocol nodes after enablement of protocols

mmces address list			
Address	Node	Group	Attribute
10.11.80.54	protocol-node-2	none	none
10.11.80.55	protocol-node-1	none	none
10.30.16.5	protocol-node-1	none	none
10.30.16.6	protocol-node-2	none	none
10.30.16.7	protocol-node-1	none	none
10.30.16.8	protocol-node-2	none	none
10.30.17.100	protocol-node-1	none	none
10.30.17.101	protocol-node-2	none	none
10.30.17.102	protocol-node-2	none	none
object_database_node,object_singleton_node			
10.30.17.103	protocol-node-1	none	none

Related concepts

[Setting up Cluster Export Services shared root file system](#)

If a shared root file system through the installer is not set up, you must create one for Cluster Export Services (CES).

[Configuring Cluster Export Services nodes](#)

If you do not configure Cluster Export Services (CES) nodes through the installer, you must configure them before you configure any protocols.

[Configuring CES protocol service IP addresses](#)

Protocol services are made available through Cluster Export Services (CES) protocol service IP addresses. These addresses are separate from the IP addresses that are used internally by the cluster.

[Deploying Cluster Export Services packages on existing IBM Storage Scale nodes](#)

Use the following instructions to copy packages on your protocol nodes and to deploy these packages.

1. Copy the required packages to the protocol node from the location where the self-extracting package was extracted.

By default, installation images are extracted to the target directory /usr/lpp/mmfs/5.2.1.x.
2. Install packages by issuing the following command:

```
rpm -ivh Package_Name1 Package_Name2 ... Package_NameN
```

For a list of packages applicable for the current IBM Storage Scale release, see *Manually installing the software packages on Linux nodes in IBM Storage Scale: Concepts, Planning, and Installation Guide*.

3. Set the server licenses for each CES node by issuing the following command:

```
mmchlicense server --accept -N CESNodeIPs
```

For example,

```
mmchlicense server --accept -N 203.0.113.7,203.0.113.9
```

4. Enable CES by issuing the following command:

```
mmchnode -N ces_nodes --ces-enable
```

For example,

```
mmchnode -N 203.0.113.7,203.0.113.9 --ces-enable
```

5. Assign export IP addresses for each export IP by issuing the following command:

```
mmces address add --ces-ip export_IP
```

Related concepts

[Setting up Cluster Export Services shared root file system](#)

If a shared root file system through the installer is not set up, you must create one for Cluster Export Services (CES).

[Configuring Cluster Export Services nodes](#)

If you do not configure Cluster Export Services (CES) nodes through the installer, you must configure them before you configure any protocols.

[Configuring CES protocol service IP addresses](#)

Protocol services are made available through Cluster Export Services (CES) protocol service IP addresses. These addresses are separate from the IP addresses that are used internally by the cluster.

[CES IP aliasing to network adapters on protocol nodes](#)

Cluster Export Services (CES) is a functionality in IBM Storage Scale that enables NFS and SMB protocols. Irrespective of which protocols you choose, all are accessible through a floating pool of IP addresses called CES IP addresses. This pool of CES IP addresses is considered floating because each IP can move independently among all protocol nodes. During a protocol node failure, accessibility to all protocols is maintained as the CES IP addresses automatically move from the failed protocol node to a healthy protocol node. Use this information to understand how CES IP addresses are assigned and are aliased to adapters with or without VLAN tagging.

[Verifying the final CES configurations](#)

After you finish configuring the Cluster Export Services (CES), you must verify the final configuration.

Verifying the final CES configurations

After you finish configuring the Cluster Export Services (CES), you must verify the final configuration.

To verify your configuration, run the following command:

```
mmlscluster --ces
```

For more information about **mmces node list** and **mmces address list**, see the topic *mmces command* in *IBM Storage Scale: Command and Programming Reference Guide*.

For more information about configuring and enabling SMB and NFS services, see [“Configuring and enabling SMB, S3, and NFS protocol services” on page 287](#).

Related concepts

[Setting up Cluster Export Services shared root file system](#)

If a shared root file system through the installer is not set up, you must create one for Cluster Export Services (CES).

[Configuring Cluster Export Services nodes](#)

If you do not configure Cluster Export Services (CES) nodes through the installer, you must configure them before you configure any protocols.

[Configuring CES protocol service IP addresses](#)

Protocol services are made available through Cluster Export Services (CES) protocol service IP addresses. These addresses are separate from the IP addresses that are used internally by the cluster.

CES IP aliasing to network adapters on protocol nodes

Cluster Export Services (CES) is a functionality in IBM Storage Scale that enables NFS and SMB protocols. Irrespective of which protocols you choose, all are accessible through a floating pool of IP addresses called CES IP addresses. This pool of CES IP addresses is considered floating because each IP can move independently among all protocol nodes. During a protocol node failure, accessibility to all protocols is maintained as the CES IP addresses automatically move from the failed protocol node to a healthy protocol node. Use this information to understand how CES IP addresses are assigned and are aliased to adapters with or without VLAN tagging.

Deploying Cluster Export Services packages on existing IBM Storage Scale nodes

Use the following instructions to copy packages on your protocol nodes and to deploy these packages.

Creating and configuring file systems and filesets for exports

If you have not done so, create the file systems and the filesets for the data to be exported through the protocol services. For more information, see `mmcrfs` and `mmcrfileset` in *IBM Storage Scale: Command and Programming Reference Guide*.

Creating a fileset through the GPFS GUI

To create a fileset, log on to the IBM Storage Scale GUI and select **Files > Filesets > Create Fileset**.

When the file system is intended for CES export, IBM strongly recommends that you configure the file systems to allow only NFSv4 ACLs through the **-k nfs4** option for `mmcrfs`. For using SMB and NFS CES protocol access, configuring the file system with **-k nfs4** is required. When you use the default configuration profiles (`/usr/lpp/mmfs/profiles`) that are included with IBM Storage Scale, the NFSv4 ACL setting is already set from the profile configuration (see “[Authorizing file protocol users](#)” on page 475 for details). Also, if quotas must be used, enable the quota usage during the file system creation.

If you are using HDFS, see the multi-protocol limitation under *Hadoop ACL and IBM Storage Scale protocols* in [IBM Storage Scale Big Data and Analytics Support documentation](#).

Note: Ensure that all GPFS file systems used to export data via NFS are mounted with the `syncnfs` option in order to prevent clients from running into data integrity issues during failover. It is recommended to use the `mmchfs` command to set the `syncnfs` option as default when you mount the GPFS file system.

For more information on creating protocol data exports, see *File system considerations for the NFS protocol* and *Fileset considerations for creating protocol data exports* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Configuring with the installation toolkit

You can use the configuration options of the installation toolkit to configure GPFS and protocols on an ongoing basis, as an alternative to the other GPFS cluster creation and configuration commands.

For detailed information about using the installation toolkit to configure GPFS and protocols, see the following:

- **`spectrumscale` command** in *IBM Storage Scale: Command and Programming Reference Guide*.
- *Installing IBM Storage Scale on Linux nodes and deploying protocols* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
- *Using the spectrumscale installation toolkit to perform installation tasks: Explanations and examples* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Deleting a Cluster Export Services node from an IBM Storage Scale cluster

Use this information to delete a Cluster Export Services (CES) node from an IBM Storage Scale cluster.

1. On a node other than the one you want to delete from the cluster, issue the following command to suspend the node:

```
# mmces node suspend -N <Node_to_Delete>
```

2. On the node that you want to delete from the cluster, issue the following commands to stop the CES services:

```
# mmces service stop nfs  
# mmces service stop smb  
# mmces service stop s3
```

In this example, it is assumed that all three protocols are enabled on the node that you want to delete from the cluster.

3. On a node other than the one you want to delete from the cluster, issue the following command to disable CES on the node:

```
# mmchnode -N <Node_to_Delete> --ces-disable
```

4. On a node other than the one you want to delete from the cluster, issue the following command to shut down GPFS on the node:

```
# mmshutdown -N <Node_to_Delete>
```

5. On a node other than the one you want to delete from the cluster, issue the following command to delete the node from the cluster:

```
# mmdeinode -N <Node_to_Delete>
```

6. If you disabled file audit logging in step 1, you can enable it by following the instructions in [“Enabling file audit logging on a file system” on page 125](#).

Setting up Cluster Export Services groups in an IBM Storage Scale cluster

After IBM Storage Scale is successfully installed and deployed, you can set up Cluster Export Services (CES) groups that are specific to nodes and CES IPs on a cluster that is working correctly by using the following information.

1. Set the CES nodes in the cluster to the corresponding groups by issuing the **mmchnode --ces-group** command. For example:

```
mmchnode --ces-group Site1 -N prt001st001  
mmchnode --ces-group Site1 -N prt002st001  
mmchnode --ces-group Site2 -N prt003st001  
mmchnode --ces-group Site2 -N prt004st001
```

Note: CES group names are not case-sensitive.

In the example, protocol nodes prt001st001 and prt002st001 are set to the Site1 CES group, and protocol nodes prt003st001 and prt004st001 are set to the site2 CES group.

2. Assign CES IPs to the corresponding CES groups by issuing the **mmces address change** command. For example:

```
mmces address change --ces-ip 192.0.2.20,192.0.2.21,192.0.2.22,192.0.2.23  
--ces-group Site1
```

```
mmces address change --ces-ip 192.0.3.20,192.0.3.21,192.0.3.22  
--ces-group Site2
```

3. To verify the CES groups your nodes belong to, issue the **mmces node list command**.

The system displays information similar to this:

Node Name	Node Flags	Node Groups
10 prt005st001	none	site2
11 prt006st001	none	site2
12 prt007st001	none	site2
13 prt008st001	none	site2
6 prt001st001	none	site1
7 prt002st001	none	site1
8 prt003st001	none	site1
9 prt004st001	none	site1

4. To verify the groups your CES IPs belong to, issue the **mmces address list command**. The system displays information similar to the following:

Address	Node	Group	Attribute
10.18.52.30	prt001st001	site1	object_singleton_node, object_database_node
10.18.52.31	prt002st001	site1	none
10.18.52.32	prt003st001	site1	none
10.18.52.33	prt004st001	site1	none
10.18.60.30	prt005st001	site2	none
10.18.60.31	prt006st001	site2	none
10.18.60.32	prt007st001	site2	none
10.18.60.33	prt008st001	site2	none

Setting up self-signed SSL/TLS certificates for secure communication between the S3 client and the S3 service

1. Create a Subject Alternative Name (SAN) config file.

```
# san.cnf
#
[req]
req_extensions = req_ext
distinguished_name = req_distinguished_name
#
[req_distinguished_name]
CN = localhost
#
[req_ext]
# The subjectAltName line directly specifies the domain names and IP addresses that the
certificate should be valid for.
# This ensures the SSL certificate matches the domain or IP used in your S3 command.
#
# Example:
# 'DNS:localhost' makes the certificate valid when accessing S3 storage via 'localhost'.
# 'DNS:cess3-domain-name-example.com' adds a specific domain to the certificate. Replace
'cess3-domain-name-example.com' with your actual domain.
# 'IP:<nsfs-server-ip>' includes an IP address. Replace '<nsfs-server-ip>' with the actual
IP address of your S3 server.
subjectAltName = DNS:localhost,DNS:cess3-domain-name-example.com,IP:<nsfs-server-ip>
```

where:

san.cnf

This file specifies the domain names or IP addresses that will be included in the SSL certificate.

CN

The Common Name (CN) sets the primary domain for the certificate, and additional domains or IPs can be listed under subjectAltName.

Replace placeholders such as `cess3-domain-name-example.com` and with your actual domain and IP address.

2. Generate TLS Key, CSR, and CRT Files by using OpenSSL.

Generate the necessary TLS key (`tls.key`), certificate signing request (`tls.csr`), and SSL certificate (`tls.crt`) files for secure communication between the S3 client and the S3 service.

```
$ sudo openssl genpkey -algorithm RSA -out tls.key
$ sudo openssl req -new -key tls.key -out tls.csr -config san.cnf -subj "/CN=localhost"
$ sudo openssl x509 -req -days 365 -in tls.csr -signkey tls.key -out tls.crt -extfile
san.cnf -extensions req_ext
```

3. Move `tls.key` and `tls.crt` under `{cesSharedRoot_path}/ces/s3-config/certificates`.

```
# get cesSharedRoot_path by running at a CES node - `mmlsconfig | grep cesSharedRoot'
$ sudo mv tls.key {cesSharedRoot_path}/ces/s3-config/certificates/
$ sudo mv tls.crt {cesSharedRoot_path}/ces/s3-config/certificates/
```

4. Restart the S3 service on CES nodes.

```
# mmces service stop s3-a
# mmces service start s3-a
```

5. Create S3 CLI alias by including `tls.crt` using `AWS_CA_BUNDLE=/path/to/tls.crt`.

- Ensure to replace credentials placeholders with their respective values, and the placeholder either with localhost or the domain name or IP of the node which is running the S3 service.

```
alias s3_ssl='AWS_CA_BUNDLE= {cesSharedRoot_path}/ces/s3-config/certificates/
tls.crt AWS_ACCESS_KEY_ID=add_your_access_key AWS_SECRET_ACCESS_KEY=add_your_secret_key
aws --endpoint https://<endpoint>:6443 s3'
```

6. Copy the `tls.crt` to the client, where AWS CLI commands are being run. Then, run the following command:

```
alias s3_ssl='AWS_ACCESS_KEY_ID=add_your_access_key
AWS_SECRET_ACCESS_KEY=add_your_secret_key aws --endpoint https://<endpoint>:6443 s3 --ca-
bundle /root/tls.crt '
```

7. Try to run an S3 list of buckets by using the `s3` alias.

```
s3_ssl ls
```

Chapter 4. Configuring and tuning your system for GPFS

In addition to configuring your GPFS cluster, you need to configure and tune your system.

For more information, see *GPFS cluster creation considerations* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Values suggested here reflect evaluations made at the time this documentation was written. For the latest system configuration and tuning settings, see “[General system configuration and tuning considerations](#)” on page 65.

Additional GPFS and system configuration and tuning considerations include:

- “[General system configuration and tuning considerations](#)” on page 65
- “[Linux configuration and tuning considerations](#)” on page 70
- “[AIX configuration and tuning considerations](#)” on page 72

For information on installing and configuring Windows on systems that will be added to a GPFS cluster, see *Configuring Windows in IBM Storage Scale: Concepts, Planning, and Installation Guide*.

For more information on using multiple token servers, see “[Using multiple token servers](#)” on page 981.

General system configuration and tuning considerations

You must consider some general system configuration and tuning considerations. This topic points you to the detailed information.

For the latest system configuration settings, see the [IBM Storage Scale FAQ](#) in IBM Documentation.

Configuration and tuning considerations for all systems include:

- “[Clock synchronization](#)” on page 65
- “[GPFS administration security](#)” on page 66
- “[Cache usage](#)” on page 66
- Chapter 5, “[Parameters for performance tuning and optimization](#),” on page 75
- “[Access patterns](#)” on page 69
- “[Aggregate network interfaces](#)” on page 69
- “[Swap space](#)” on page 70

Clock synchronization

The clocks of all nodes in the GPFS cluster must be synchronized. If this is not done, NFS access to the data and other GPFS file system operations may be disrupted.

Related concepts

[GPFS administration security](#)

Before administering your GPFS file system, make certain that your system has been properly configured for security.

[Cache usage](#)

GPFS creates a number of cache segments on each node in the cluster. The amount of cache is controlled by three attributes.

[Access patterns](#)

GPFS attempts to recognize the pattern of accesses (such as strided sequential access) that an application makes to an open file. If GPFS recognizes the access pattern, it will optimize its own behavior.

Aggregate network interfaces

It is possible to aggregate multiple physical Ethernet interfaces into a single virtual interface. This is known as *Channel Bonding* on Linux and *EtherChannel/IEEE 802.3ad Link Aggregation* on AIX.

Swap space

It is important to configure a swap space that is large enough for the needs of the system.

GPFS administration security

Before administering your GPFS file system, make certain that your system has been properly configured for security.

This includes:

- Assigning root authority to perform all GPFS administration tasks except:
 - Tasks with functions limited to listing GPFS operating characteristics.
 - Tasks related to modifying individual user file attributes.
- Establishing the authentication method between nodes in the GPFS cluster.:
 - Until you set the authentication method, you cannot issue any GPFS commands.
- Designating a remote communication program for remote shell and remote file copy commands:
 - The default remote communication commands are `scp` and `ssh`. You can designate any other remote commands if they have the same syntax.
 - Regardless of which remote commands have been selected, the nodes that you plan to use for administering GPFS must be able to execute commands on any other node in the cluster without the use of a password and without producing any extraneous messages.

Related concepts

Cache usage

GPFS creates a number of cache segments on each node in the cluster. The amount of cache is controlled by three attributes.

Access patterns

GPFS attempts to recognize the pattern of accesses (such as strided sequential access) that an application makes to an open file. If GPFS recognizes the access pattern, it will optimize its own behavior.

Aggregate network interfaces

It is possible to aggregate multiple physical Ethernet interfaces into a single virtual interface. This is known as *Channel Bonding* on Linux and *EtherChannel/IEEE 802.3ad Link Aggregation* on AIX.

Swap space

It is important to configure a swap space that is large enough for the needs of the system.

Related tasks

Clock synchronization

The clocks of all nodes in the GPFS cluster must be synchronized. If this is not done, NFS access to the data and other GPFS file system operations may be disrupted.

Cache usage

GPFS creates a number of cache segments on each node in the cluster. The amount of cache is controlled by three attributes.

These attributes have default values at cluster creation time and might be changed through the **mmchconfig** command:

pagepool

The GPFS pagepool attribute is used to cache user data and file system metadata. The pagepool mechanism allows GPFS to implement read and write requests asynchronously. Increasing the size

of the pagepool attribute increases the amount of data or metadata that GPFS can cache without requiring synchronous I/O. The operating system and other software that is running on the node might restrict the amount of memory available for GPFS on a particular node.

The optimal size of the pagepool attribute depends on the needs of the application and effective caching of its reaccessed data. For systems where applications access large files, reuse data, benefit from GPFS prefetching of data, or have a random I/O pattern, increasing the value for the pagepool attribute might prove beneficial. However, if the value is set too large, GPFS starts with the maximum that the system allows. See the GPFS log for the value it is running at.

To change the size of the pagepool attribute to 4 GB:

```
mmchconfig pagepool=4G
```

maxFilesToCache

The total number of different files that can be cached at one time. Every entry in the file cache requires some pageable memory to hold the content of the file's inode plus control data structures. This is in addition to any of the file's data and indirect blocks that might be cached in the page pool.

While the total amount of memory that is required for inodes, attributes and control data structures varies based on the functions that are being used, it can be estimated as a maximum of 10 KB per file that is cached.

Valid values of maxFilesToCache range from 1 through 100,000,000. For systems where the applications use many files, of any size, increasing the value for maxFilesToCache might prove beneficial. This is true for systems where many small files are accessed. The value must be large enough to handle the number of concurrently open files plus allow caching of recently used files.

If the user does not specify a value for maxFilesToCache, the default value is 4000.

Note: For CES nodes where applications use more than 250,000 files, double the value of maxFilesToCache to the files used. For example, for 250,000 files, maxFilesToCache value is recommended to be 500,000.

maxStatCache

This parameter sets aside extra pageable memory to cache attributes of files that are not currently in the regular file cache. This is useful to improve the performance of both the system and GPFS stat() calls for applications with a working set that does not fit in the regular file cache. For systems where applications test the existence of files, or the properties of files without opening them, as backup applications do, increasing the value for maxStatCache can improve performance.

The memory that is occupied by the stat cache can be calculated as:

```
maxStatCache × 480 bytes
```

The valid range for **maxStatCache** is 0 - 100,000,000. If you do not specify values for **maxFilesToCache** and **maxStatCache**, the default value of **maxFilesToCache** is 4000 and the default value of **maxStatCache** is 1000. If you specify a value for **maxFilesToCache** but not for **maxStatCache**, the default value of **maxStatCache** is $4 * \text{maxFilesToCache}$ or 10000, whichever is smaller.

Note: For improving directory listing performance on CES nodes, double the maxStatCache values to that of maxFilesToCache. For example, for a maxFilesToCache value of 500,000, maxStatCache value is recommended to be 1,000,000.

The total amount of memory GPFS uses to cache file data and metadata is arrived at by adding pagepool to the amount of memory that is required to hold inodes and control data structures (**maxFilesToCache** × 10 KB), and the memory for the stat cache (**maxStatCache** × 480 bytes) together. The combined amount of memory to hold inodes, control data structures, and the stat cache is limited to 50% of the physical memory on a node that is running GPFS.

During configuration, you can specify the **maxFilesToCache**, **maxStatCache**, and **pagepool** attributes that control how much cache is dedicated to GPFS. These values can be changed later, so experiment

with larger values to find the optimum cache size that improves GPFS performance without negatively affecting other applications.

The **mmchconfig** command can be used to change the values of `maxFilesToCache`, `maxStatCache`, and `pagepool`. The `pagepool` parameter is the only one of these parameters that might be changed while the GPFS daemon is running. A change to the `pagepool` attribute occurs immediately when you are using the `-i` option on the **mmchconfig** command. Changes to the other values are effective only after the daemon is restarted.

For more information on these cache settings for GPFS, see *GPFS and memory in IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Related concepts

[GPFS administration security](#)

Before administering your GPFS file system, make certain that your system has been properly configured for security.

[Access patterns](#)

GPFS attempts to recognize the pattern of accesses (such as strided sequential access) that an application makes to an open file. If GPFS recognizes the access pattern, it will optimize its own behavior.

[Aggregate network interfaces](#)

It is possible to aggregate multiple physical Ethernet interfaces into a single virtual interface. This is known as *Channel Bonding* on Linux and *EtherChannel/IEEE 802.3ad Link Aggregation* on AIX.

[Swap space](#)

It is important to configure a swap space that is large enough for the needs of the system.

Related tasks

[Clock synchronization](#)

The clocks of all nodes in the GPFS cluster must be synchronized. If this is not done, NFS access to the data and other GPFS file system operations may be disrupted.

The GPFS token system's effect on cache settings

Lock tokens play a role in maintaining cache consistency between nodes.

A token allows a node to cache data it has read from disk, because the data cannot be modified elsewhere without revoking the token first.

Note the following facts about the attributes `maxFilesToCache` and `maxStatCache`:

- For the default values, see *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

In versions of IBM Storage Scale earlier than 5.0.2, the `maxStatCache` attribute is not effective on the Linux platform unless the Local Read-Only Cache (LROC) is configured. For more information, see *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

- The `maxStatCache` attribute can be set higher on user-interactive nodes and lower on dedicated compute nodes, because `ls -l` performance is mostly a human response issue.
- The `maxFilesToCache` attribute must be large enough to handle the number of concurrently open files and allow caching of recently used files. Note that increasing this value increases the memory that is used by IBM Storage Scale. For information about calculating the memory that is consumed by `maxFilesToCache` attribute, see [“Cache usage” on page 66](#).
- The attributes `maxFilesToCache` and `maxStatCache` are indirectly affected by the number of manager nodes that are defined in the cluster. Having more manager nodes typically allows more tokens to be managed by the cluster.

Access patterns

GPFS attempts to recognize the pattern of accesses (such as strided sequential access) that an application makes to an open file. If GPFS recognizes the access pattern, it will optimize its own behavior.

For example, GPFS can recognize sequential reads and will retrieve file blocks before they are required by the application. However, in some cases GPFS does not recognize the access pattern of the application or cannot optimize its data transfers. In these situations, you may improve GPFS performance if the application explicitly discloses aspects of its access pattern to GPFS through the `gpfs_fcntl()` library call.

Related concepts

GPFS administration security

Before administering your GPFS file system, make certain that your system has been properly configured for security.

Cache usage

GPFS creates a number of cache segments on each node in the cluster. The amount of cache is controlled by three attributes.

Aggregate network interfaces

It is possible to aggregate multiple physical Ethernet interfaces into a single virtual interface. This is known as *Channel Bonding* on Linux and *EtherChannel/IEEE 802.3ad Link Aggregation* on AIX.

Swap space

It is important to configure a swap space that is large enough for the needs of the system.

Related tasks

Clock synchronization

The clocks of all nodes in the GPFS cluster must be synchronized. If this is not done, NFS access to the data and other GPFS file system operations may be disrupted.

Aggregate network interfaces

It is possible to aggregate multiple physical Ethernet interfaces into a single virtual interface. This is known as *Channel Bonding* on Linux and *EtherChannel/IEEE 802.3ad Link Aggregation* on AIX.

GPFS supports by using such aggregate interfaces. The main benefit is increased bandwidth. The aggregated interface has the network bandwidth close to the total bandwidth of all its physical adapters. Another benefit is improved fault tolerance. If a physical adapter fails, the packets are automatically sent on the next available adapter without service disruption.

EtherChannel and IEEE802.3ad each requires support within the Ethernet switch. Refer to the product documentation for your switch to determine if EtherChannel is supported.

For details on how to configure EtherChannel and IEEE 802.3ad Link Aggregation and verify whether the adapter and the switch are operating with the correct protocols for IEEE 802.3ad, consult the operating system documentation.

Hint: Make certain that the switch ports are configured for **LACP** (the default is **PAGP**).

For additional service updates regarding the use of EtherChannel:

1. Go to the [IBM Support Portal \(www.ibm.com/support\)](http://www.ibm.com/support).
2. In the **Search** box, enter the search term *EtherChannel*.
3. Click **Search**.

Hint: A useful command for troubleshooting, where device is the Link Aggregation device, is:

```
entstat -d device
```

Related concepts

GPFS administration security

Before administering your GPFS file system, make certain that your system has been properly configured for security.

Cache usage

GPFS creates a number of cache segments on each node in the cluster. The amount of cache is controlled by three attributes.

Access patterns

GPFS attempts to recognize the pattern of accesses (such as strided sequential access) that an application makes to an open file. If GPFS recognizes the access pattern, it will optimize its own behavior.

Swap space

It is important to configure a swap space that is large enough for the needs of the system.

Related tasks

Clock synchronization

The clocks of all nodes in the GPFS cluster must be synchronized. If this is not done, NFS access to the data and other GPFS file system operations may be disrupted.

Swap space

It is important to configure a swap space that is large enough for the needs of the system.

While the actual configuration decisions should be made when considering the memory requirements of other applications, it is a good practice to configure at least as much swap space as there is physical memory on a given node.

Related concepts

GPFS administration security

Before administering your GPFS file system, make certain that your system has been properly configured for security.

Cache usage

GPFS creates a number of cache segments on each node in the cluster. The amount of cache is controlled by three attributes.

Access patterns

GPFS attempts to recognize the pattern of accesses (such as strided sequential access) that an application makes to an open file. If GPFS recognizes the access pattern, it will optimize its own behavior.

Aggregate network interfaces

It is possible to aggregate multiple physical Ethernet interfaces into a single virtual interface. This is known as *Channel Bonding* on Linux and *EtherChannel/IEEE 802.3ad Link Aggregation* on AIX.

Related tasks

Clock synchronization

The clocks of all nodes in the GPFS cluster must be synchronized. If this is not done, NFS access to the data and other GPFS file system operations may be disrupted.

Linux configuration and tuning considerations

Configuration and tuning considerations for the Linux nodes in your system include the use of the updatedb utility, the `vm.min_free_kbytes` kernel tunable, and several other options that can improve GPFS performance.

For the latest system configuration and tuning settings, see “[General system configuration and tuning considerations](#)” on page 65.

For more configuration and tuning considerations for Linux nodes, see the following topics:

1. [“updatedb considerations” on page 71](#)
2. [“Memory considerations” on page 71](#)
3. [“GPFS helper threads” on page 71](#)

4. [“Communications I/O” on page 71](#)
5. [“Disk I/O” on page 72](#)

updatedb considerations

On some Linux distributions, the system is configured by default to run the file system indexing utility updatedb through the cron daemon on a periodic basis (usually daily).

This utility traverses the file hierarchy and generates a large I/O load. For this reason, it is configured by default to skip certain file system types and nonessential file systems. However, the default configuration does not prevent updatedb from traversing GPFS file systems. In a cluster this results in multiple instances of updatedb traversing the same GPFS file system simultaneously. This causes general file system activity and lock contention in proportion to the number of nodes in the cluster. On smaller clusters, this may result in a relatively short-lived spike of activity, while on larger clusters, depending on the overall system throughput capability, the period of heavy load may last longer. Usually the file system manager node will be the busiest, and GPFS would appear sluggish on all nodes. Re-configuring the system to either make updatedb skip all GPFS file systems or only index GPFS files on one node in the cluster is necessary to avoid this problem.

Memory considerations

It is recommended that you adjust the `vm.min_free_kbytes` kernel tunable. This tunable controls the amount of free memory that Linux kernel keeps available (that is, not used in any kernel caches).

When `vm.min_free_kbytes` is set to its default value, on some configurations it is possible to encounter memory exhaustion symptoms when free memory should in fact be available. Setting `vm.min_free_kbytes` to 5 – 6% of the total amount of physical memory, but no more than 2 GB, can prevent this problem.

GPFS helper threads

GPFS uses helper threads such as `prefetchThreads` and `workerThreads` to improve performance.

Since systems vary, it is suggested you simulate an expected workload in GPFS and examine available performance indicators on your system. For instance, some SCSI drivers publish statistics in the `/proc/scsi` directory. If your disk driver statistics indicate that there are many *queued requests*, then it might mean you should throttle back the helper threads in GPFS.

For more information, see [Chapter 5, “Parameters for performance tuning and optimization,” on page 75](#).

Communications I/O

Values suggested here reflect evaluations made at the time this documentation was written. For the latest system configuration and tuning settings, see [Chapter 4, “Configuring and tuning your system for GPFS,” on page 65](#).

To optimize the performance of GPFS and your network, it is suggested you do the following:

- Enable Jumbo Frames if your switch supports it.

If GPFS is configured to operate over Gigabit Ethernet, set the MTU size for the communication adapter to 9000.

- Verify `/proc/sys/net/ipv4/tcp_window_scaling` is enabled. It should be by default.
- Tune the TCP window settings by adding these lines to the `/etc/sysctl.conf` file:

```
# increase Linux TCP buffer limits
net.core.rmem_max = 8388608
net.core.wmem_max = 8388608
# increase default and maximum Linux TCP buffer sizes
net.ipv4.tcp_rmem = 4096 262144 8388608
net.ipv4.tcp_wmem = 4096 262144 8388608
```

After these changes are made to the `/etc/sysctl.conf` file, apply the changes to your system:

1. Issue the `sysctl -p /etc/sysctl.conf` command to set the kernel settings.
2. Issue the `mmshutdown -a` command and then issue `mmstartup -a` command to restart GPFS

Disk I/O

To optimize disk I/O performance, you should consider the following options for NSD servers or other GPFS nodes that are directly attached to a SAN over a Fibre Channel (FC) network:

1. The storage server cache settings can impact GPFS performance if not set correctly.
2. When the storage server disks are configured for RAID5, some configuration settings can affect GPFS performance. These settings include:
 - GPFS block size
 - Maximum I/O size of the Fibre Channel host bus adapter (HBA) device driver
 - Storage server RAID5 stripe size
3. These suggestions may avoid the performance penalty of read-modify-write at the storage server for GPFS writes. Examples of the suggested settings are:
 - 8+P RAID5
 - GPFS block size = 512K
 - Storage Server RAID5 segment size = 64K (RAID5 stripe size=512K)
 - Maximum IO size of FC HBA device driver = 512K
 - 4+P RAID5
 - GPFS block size = 256K
 - Storage Server RAID5 segment size = 64K (RAID5 stripe size = 256K)
 - Maximum IO size of FC HBA device driver = 256K

For the example settings using 8+P and 4+P RAID5, the RAID5 parity can be calculated from the data written and will avoid reading from disk to calculate the RAID5 parity. The maximum IO size of the FC HBA device driver can be verified using `iostat` or the Storage Server performance monitor. In some cases, the device driver may need to be patched to increase the default maximum IO size.

4. The GPFS parameter `maxMBpS` can limit the maximum throughput of an NSD server or a single GPFS node that is directly attached to the SAN with a FC HBA. The default value is 2048. The `maxMBpS` parameter is changed by issuing the `mmchconfig` command. If this value is changed, restart GPFS on the nodes, and test the read and write performance of a single node and a large number of nodes.

AIX configuration and tuning considerations

For the latest system configuration settings, see the [IBM Storage Scale FAQ in IBM Documentation](#).

GPFS use with Oracle

When GPFS is used with Oracle, the configuration and tuning include the considerations that are mentioned in this section.

- While setting up your LUNs, it is important to create the NSD, such that they map one-to-one with a LUN that is a single RAID device.
- For file systems that are holding large Oracle databases, set the GPFS file system block size through the `mmcrfs` command by using the `-B` option, to a larger value:
 - 512 KB is generally suggested.

- 256 KB is suggested when there is activity other than Oracle by using the file system and many small files exist, which are not in the database.
- 1 MB is suggested for file systems that are 100 TB or larger in size.

The large block size makes the allocation of space for the databases manageable and has no effect on performance when Oracle is using the Asynchronous I/O (AIO) and Direct I/O (DIO) features of AIX.

- Set the GPFS worker threads through the `mmchconfig worker1Threads` command to allow the maximum parallelism of the Oracle AIO threads:
 - Adjust the GPFS prefetch threads accordingly through the `mmchconfig prefetchThreads` command. The maximum value of `prefetchThreads` plus `worker1Threads` plus `nsdMaxWorkerThreads` is 8192 on all 64-bit platforms.
 - When requiring GPFS sequential I/O, set the prefetch threads in the range 50 - 100 (the default is 72).

Note: These changes through the `mmchconfig` command take effect upon restart of the GPFS daemon.

- The number of AIX AIO kprocs to create is approximately the same as the GPFS `worker1Threads` setting.
- The AIX AIO `maxservers` setting is the number of kprocs PER CPU. It is suggested to set it slightly larger than the value of `worker1Threads` divided by the number of CPUs. For example, if `worker1Threads` is set to 500 on a 32-way SMP, set `maxservers` to 20.
- Set the Oracle database block size equal to the LUN segment size or a multiple of the LUN pdisk segment size.
- Set the Oracle read-ahead value to prefetch one or two full GPFS blocks. For example, if your GPFS block size is 512 KB, set the Oracle blocks to either 32 or 64 16 KB blocks.
- Do not use the `dio` option on the `mount` command as this forces DIO when accessing all files. Oracle automatically uses DIO to open database files on GPFS.
- When running Oracle RAC 10 g, it is suggested you increase the value for `OPROCD_DEFAULT_MARGIN` to at least 500 to avoid possible random restarts of nodes.

In the control script for the Oracle CSS daemon, which is located in `/etc/init.cssd` the value for `OPROCD_DEFAULT_MARGIN` is set to 500 (milliseconds) on all UNIX derivatives except for AIX. For AIX, this value is set to 100. From a GPFS perspective, even 500 milliseconds maybe too low in situations where node failover might take up to a minute or two to resolve. However, if during node failure the surviving node is already doing direct IO to the `oprocd` control file, it must have the necessary tokens and indirect block that is cached and therefore must not wait during failover.

Chapter 5. Parameters for performance tuning and optimization

Use these parameters with the `mmchconfig` command for performance tuning and optimization.

Tuning guide for frequently changed parameters

autoload

When **autoload** is set to yes, GPFS starts automatically on the nodes that are rebooted. The rebooted nodes rejoin the cluster. The `file system automount` option is set to **yes**, and the file system is mounted. The default value of this parameter is **no**.

Important: Set **autoload** to **no** before you fix hardware issues and performing system maintenance.

deadlockDetectionThreshold

When **deadlockDetectionThreshold** is set to 0, the GPFS dead-lock detection feature is disabled. The default value of this parameter is 300 seconds.

Important: You must enable the GPFS dead-lock detection feature to collect debug data and resolve dead lock issue in a cluster. If dead-lock events occur frequently, fix the problem instead of disabling the feature.

defaultHelperNodes

The nodes that are added to **defaultHelperNodes** are used in running certain commands, such as `mmrestripefs`. Running the GPFS command on partial nodes in a cluster, such as running the `mmrestripefs` command on all NSD server nodes, might have a better performance. The default value of this parameter is all nodes in cluster.

Important: Set the `-N` option for GPFS management commands or change the value of **defaultHelperNodes** before you run the GPFS management commands.

maxFilesToCache

The **maxFilesToCache** parameter specifies the number of files that can be cached by each node. The range of valid values for **maxFilesToCache** is 1 - 100,000,000. The default value is 4000. The value of this parameter must be large enough to handle the number of concurrently open files and to allow the caching of recently used files.

Changing the value of **maxFilesToCache** affects the amount of memory that is used on the node. In a large cluster, a change in the value of **maxFilesToCache** is greatly magnified. Increasing **maxFilesToCache** in a large cluster with hundreds of nodes increases the number of tokens a token manager needs to store. Ensure that the manager node has enough memory and **tokenMemLimit** is increased when you are running GPFS version 4.1.1 and earlier. Therefore, increasing the value of **maxFilesToCache** on large clusters usually happens on a subset of nodes that are used as log-in nodes, SMB and NFS exporters, email servers, and other file servers.

For systems on which applications use many files, increasing the value of **maxFilesToCache** might be beneficial, especially where many small files are accessed.



Trouble: Setting the **maxFilesToCache** parameter to a high value results in a large amount of memory that is being allocated for internal data buffering. If the value of **maxFilesToCache** is set too high, some operations in IBM Storage Scale might not have enough memory to run in. If you set **maxFilesToCache** to a high value, then an error message might appear in the `mmfs.log` indicating that there is insufficient memory to perform an operation. To rectify, the error, try to lower the value of **maxFilesToCache**.

maxBlockSize

The value of **maxBlockSize** must be equal to or larger than the maximum block size of all the file systems in the local and remote clusters. Before you change this parameter, ensure that the GPFS daemon on each node in the cluster is shut down. The default value is 4 MB.

Note: When you migrate a cluster from an earlier version to version 5.0.0 or later, the value of maxblocksize stays the same. However, if maxblocksize was set to DEFAULT in the earlier version of the cluster, then migrating it to version 5.0.0 or later sets it explicitly to 1 MiB, that is, its default size in earlier versions. To change maxBlockSize to the default size after you migrate to version 5.0.0 or later, set maxblocksize=DEFAULT (4 MiB).

For more information, see the *mmcrfs* and *mmchconfig* commands in the *IBM Storage Scale: Command and Programming Reference Guide*.

maxMBps

The **maxMBps** parameter indicates the maximum throughput in megabytes per second that GPFS can submit into or out of a single node. GPFS calculates from this variable how many prefetch or writebehind threads to schedule for sequential file access.

In GPFS version 3.5 and later, the default value is 2048. But if the node has faster interconnect, such as InfiniBand or 40 GigE or multiple links, you can set the parameter to a higher value. As a rule, try setting **maxMBps** to twice the I/O throughput that the node can support. For example, if the node has 1 x FDR link and the GPFS configuration parameter **verbRdma** is enabled, then the expected throughput of the node is 6000 MB/s. In this case, set **maxMBps** to 12000.

Setting **maxMBps** does not guarantee the required GPFS sequential bandwidth on the node. All the layers of the GPFS stack, including the node, the network, and the storage subsystem, must be designed and tuned to meet the I/O performance requirements.

maxStatCache

The **maxStatCache** parameter sets aside the pageable memory to cache attributes of files that are not currently in the regular file cache. This improves the performance of stat() calls for applications with a working set that does not fit in the regular file cache. For systems where applications test the existence of files, or the properties of files, without opening them as backup applications do, increasing the value for **maxStatCache** can be beneficial.

For information about the default values of **maxFilesToCache** and **maxStatCache**, see the description of the **maxStatCache** attribute in the topic *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

In versions of IBM Storage Scale earlier than 5.0.2, the stat cache is not effective on the Linux platform unless the Local Read-Only Cache (LROC) is configured. For more information, see the topic *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

nsdMaxWorkerThreads

NSD server tuning. For more information about **nsdMaxWorkerThreads**, see *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

pagepool

The **pagepool** parameter is used to change the size of the data cache on each node. The default value is either one-third of the physical memory of the node or 4 GiB, whichever is smaller. The default value applies to new clusters that are installed with IBM Storage Scale 5.2.0 or higher. Otherwise, the existing default value is used. On upgrades, the existing default value is maintained.

The maximum GPFS **pagepool** size depends on the value of the **pagepoolMaxPhysMemPct** parameter and the amount of physical memory on the node. Unlike local file systems that use the operating system page cache to cache file data, GPFS allocates its own cache that is called the page pool. The GPFS page pool is used to cache user file data and file system metadata. Along with file data, the page pool supplies memory for various types of buffers such as prefetch and write behind. The default page pool size might be sufficient for sequential IO workloads. The default page pool size might not be sufficient for Random IO or workloads that involve multiple small files.

In some cases, allocating 4 GB, 8 GB, or more memory can improve the workload performance. For database applications that use Direct IO, the page pool is not used for any user data. The main purpose in this case is for system metadata and caching the indirect blocks for the files. For NSD server, if no applications or file system manager services are running on NSD server, the page pool is only used transiently by the NSD worker threads to gather data from client nodes and write the data to disk. The NSD server does not cache any of the data.

readReplicaPolicy

The **readReplicaPolicy** parameter specifies the location from which the disk must read the replicas. The valid values are default, local and fastest. The default value is default.

By default, GPFS reads the first replica even when there is no replica on the local disk. When the value of this parameter is set to local, the policy reads replicas from the local disk only if the local disk has data. An NSD server on the same subnet as the client is also considered as local. For performance considerations, this is the recommended setting for FPO environments.

When the value of this parameter is set to fastest, the policy reads replicas from the disk considering the fastest based on the read I/O statistics of the disk. In a system with SSD and regular disks, the value of **fastestPolicyCmpThreshold** can be set to a greater number, such as 100, to let GPFS refresh the slow disk speed statistics less frequently.

restripeOnDiskFailure

The **restripeOnDiskFailure** specifies whether GPFS attempts to automatically recover from certain common disk failure situations. The default value of this parameter is no.

Important: While you deploy FPO or when the HAWC feature is enabled, set the **restripeOnDiskFailure** parameter to yes.

tiebreakerDisks

For a small cluster with up to eight nodes that have SAN-attached disk systems, define all nodes as quorum nodes and use tiebreaker disks. With more than eight nodes, use only node quorum. While you are defining the tiebreaker disks, you can use the SAN-attached NSD in the file system. The default value of this parameter is null, which means no tiebreaker disk is defined.

unmountOnDiskFail

The **unmountOnDiskFail** attribute controls how the GPFS daemon responds when it detects a disk failure. For more information, see the topic *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Important:

Set the value of **unmountOnDiskFail** to **meta** in the following situations:

- FPO deployment.
- When the metadata and data replicas are more than one.

workerThreads

The **workerThreads** parameter controls an integrated group of variables that tune the file system performance in environments that are capable of high sequential and random read and write workloads and small file activity.

The default value of this parameter is 256 for a base IBM Storage Scale cluster and 512 for a cluster with protocols installed. The value for the base cluster applies to new clusters that are installed with IBM Storage Scale 5.2.0 or higher. Otherwise, the existing default value is used. A valid value can be any number in the range 1 - 8192. The -N flag is valid with this variable. This variable controls both internal and external variables. The internal variables include maximum settings for concurrent file operations, for concurrent threads that flush dirty data and metadata, and for concurrent threads that prefetch data and metadata. You can adjust the following external variables with the **mmchconfig** command:

- *logBufferCount*
- *prefetchThreads*
- *worker3Threads*

Recommendations for tuning `maxTcpConnsPerNodeConn` parameter

Starting with IBM Storage Scale 5.1.1, you can establish multiple TCP connections between nodes (with the same daemon IP address being used on each end) to optimize the use of network bandwidth. IBM

Storage Scale 5.1.5 introduces the Multi-Rail Over TCP (MROT) feature that enables the concurrent use of multiple subnets to communicate with a specified destination, and now allows the concurrent use of multiple physical network interfaces without requiring bonding to be configured. The number of connections is controlled through the **maxTcpConnsPerNodeConn** parameter, which can be changed by using the **mmchconfig** command. Valid values are 1-16, with the default being 2.

The value that is assigned to **maxTcpConnsPerNodeConn** must be defined after you consider the following factors:

- The overall bandwidth of the cluster network.
- The number of nodes in the cluster.
- With MROT, the number of physical network interfaces used for communication.
- The value that is configured for the **maxReceiverThreads** parameter.
- Memory resource implications of setting a higher value for the **maxTcpConnsPerNodeConn** parameter.

Tuning based on the overall bandwidth of the cluster network

The bandwidth that one TCP connection can achieve depends on a number of factors such as tuning, CPU, memory, and network adapter performance. However, a good starting point estimation is to assume that the bandwidth per each TCP connection can be as much as 25 Gbps. You can then set a larger value for the **maxTcpConnsPerNodeConn** parameter on faster networks so that the full network bandwidth can be used.

With MROT, if multiple network interfaces are used for communication, you can set a larger value for the **maxTcpConnsPerNodeConn** parameter, which can be greater than 8.

Note: The clusters that support RDMA for data transfers might not need to change the default value of **maxTcpConnsPerNodeConn** to improve bandwidth, as verbsRDMA is used instead of TCP for more bandwidth sensitive operations.

Tuning based on the number of nodes in the cluster

As the number of nodes in a cluster increase, you should decrease the value of the **maxTcpConnsPerNodeConn** parameter because each node requires a number of connections to other nodes, and the additional connections consume more network bandwidth. That is, large clusters with many client connections to NSD server nodes might not need to set **maxTcpConnsPerNodeConn** higher than 1, unless there are other traffic patterns such as the use of HDFS transparency, which can benefit from this tuning.

The following table provides guidelines for the value of **maxTcpConnsPerNodeConn** in proportion to the number of nodes in the cluster.

Table 4. Recommendations for selecting value for the maxTcpConnsPerNodeConn parameter		
Number of nodes in the cluster	Network bandwidth	Value of maxTcpConnsPerNodeConn
Less than 100	100 Gbps	2 - 8
100 - 1000	100 Gbps	1 - 4
1000 - 2000	100 Gbps	1 - 2
More than 2000	100 Gbps	1

For more information about the **maxTcpConnsPerNodeConn** parameter, see the *mmchconfig command* in *IBM Storage Scale: Command and Programming Reference Guide*.

Implications on receiver threads

Configuring **maxTcpConnsPerNodeConn** has a potential impact on the **maxReceiverThreads** parameter because the additional network connections might require more receiver threads. Each receiver thread can typically monitor up to 128 TCP connections, but optimal performance can be achieved when each receiver thread monitors fewer connections. The value of **maxReceiverThreads** should be selected after considering the value of the **maxTcpConnsPerNodeConn** parameter and the number of nodes in your cluster. Some large clusters need to increase the value of **maxReceiverThreads** based on the number of TCP connections that will be needed to other nodes in both local clusters and remote clusters that are joined. The total number of TCP connections that are required is calculated by using the following formula:(**maxTcpConnsPerNodeConn***(number of nodes -1))

The maximum number of receiver threads that are created on any node is defined to be the minimum of the number of logical CPUs on the node and the value of the **maxReceiverThreads** parameter. You can specify a value in the 1-128 range for the **maxReceiverThreads** parameter, with the default value being 32. For more information about how to configure **maxReceiverThreads**, see *mmchconfig command* in *IBM Storage Scale: Command and Programming Reference Guide*.

Implications on memory resources

Setting a higher value for this parameter can require more memory for allocations such as kernel socket buffers, and it can put more pressure on memory-related resources in IBM Storage Scale.

Tuning parameters change history

Parameter ¹	Added	Updated ²	Obsoleted ³
adminMode			
afmAsyncDelay		4.1	
afmAsyncOpWaitTimeout	5.0.0	5.0.1	
afmDirLookupRefreshInterval			
afmDirOpenRefreshInterval			
afmDisconnectTimeout			
afmEnableNFSSec	5.0.1		
afmEvictRange	5.2.1		
afmExpirationTimeout			
afmFastCreate	5.0.5		
afmFastLookup	5.2.0		
afmFileLookupRefreshInterval			
afmFileOpenRefreshInterval			
afmGateway	5.0.2		
afmHardMemThreshold	4.2		
afmHashVersion	4.1	5.0.0, 5.0.2, 5.0.5	
afmLookupMapSize	5.2.0		
afmMaxParallelRecoveries	5.0.0		

Parameter¹	Added	Updated²	Obsoleted³
afmNFSV4	5.2.1		
afmNFSVersion	5.1.2		
afmSyncNFSV4ACL	5.1.2		
afmNumReadThreads	4.1	4.1.0.4	
afmNumWriteThreads	4.1.0.4		
afmObjectFastReaddir	5.1.2.1		
afmObjMUCheckFName	5.2.0		
afmParallelMounts	5.0.4		
afmParallelReadChunkSize	4.1		
afmParallelReadThreshold	4.1		
afmParallelWriteChunkSize	4.1		
afmParallelWriteThreshold	4.1		
afmReadDirOnce	5.0.5		
afmReadSparseThreshold			
afmRecoveryDir	5.2.1		
afmRecoveryUseFset	5.2.1		
afmRefreshAsync	5.0.3		
afmRefreshOnce	5.0.5		
afmResyncVer2	5.1.2		
afmRevalOpWaitTimeout	5.0.0		
afmRPO	5.0.0		
afmSecondaryRW	4.2		
afmShowHomeSnapshot			
afmSyncOpWaitTimeout	5.0.0		
atimeDeferredSeconds			
autoload		4.1	
automountDir		4.1	
backgroundSpaceReclaimThreshold	5.1.0		
cesSharedRoot	4.1.1	4.2.1	
cifsBypassTraversalChecking	4.2.1		
cipherList		4.1.0.4	
cnfsGrace	4.1		
cnfsMountdPort			
cnfsNFSDprocs			
cnfsReboot	4.1		

Parameter¹	Added	Updated²	Obsoleted³
cnfsSharedRoot		4.1	
cnfsVersions	4.1		
cnfsVIP			4.0.1.4
commandAudit	4.2.1		
dataDiskCacheProtectionMethod		4.2.1	
dataDiskWaitTimeForRecovery			
dataStructureDump		4.2.1	
deadlockBreakupDelay	4.1		
deadlockDataCollectionDailyLimit	4.1	4.2, 4.2.1, 4.2.3	
deadlockDataCollectionMinInterval	4.1.1	4.2, 4.2.1	
deadlockDetectionThreshold	4.1	4.2, 4.2.1	
deadlockDetectionThresholdForShortWaiters	4.1.1		
deadlockDetectionThresholdIfOverloaded	4.1.1	4.2	4.2.1
deadlockOverloadThreshold	4.1.1	4.2, 4.2.1	
debugDataControl	4.2.1	4.2.3	
defaultHelperNodes		4.1.0.4, 5.0.1	
defaultMountDir			
disableInodeUpdateOnFdatasync			
dmapiDataEventRetry			
dmapiEventTimeout		4.2.3	
dmapiMountEvent			
dmapiMountTimeout			
dmapiSessionFailureTimeout			
enableIPv6	4.2.0	5.1.0	
enforceFilesetQuotaOnRoot			
expelDataCollectionDailyLimit	4.1.1		
expelDataCollectionMinInterval	4.1.1		
failureDetectionTime			
fastestPolicyCmpThreshold	4.1.1	4.2.1	
fastestPolicyMaxValidPeriod	4.1.1		
fastestPolicyMinDiffPercent	4.1.1		
fastestPolicyNumReadSamples	4.1.1		
fileHeatLossPercent		5.0.4	
fileHeatPeriodMinutes		5.0.4	
FIPS1402mode	4.1	4.1.0.4	

Parameter¹	Added	Updated²	Obsoleted³
frequentLeaveCountThreshold	5.0.1		
frequentLeaveTimespanMinutes	5.0.1		
ignorePrefetchLUNCount	4.2.1	4.2.3, 5.2.0	
logRecoveryThreadsPerLog	5.1.0		
logOpenParallelism	5.1.0		
logRecoveryParallelism	5.1.0		
lrocData	4.1		
lrocDataMaxFileSize	4.1		
lrocDataStubFileSize	4.1		
lrocDirectories	4.1		
lrocEnableStoringClearText	5.0.0		
lrocInodes	4.1		
maxActiveIallocSegs	5.0.2		
maxblocksize		4.1.0.4, 5.0.0	
maxBufferDescs	4.2.1		
maxDownDisksForRecovery	4.1.1		
maxFailedNodesForRecovery	4.1.1		
maxFcntlRangesPerFile			
maxFilesToCache			
maxMBpS			
maxMissedPingTimeout	4.2.1		
maxStatCache		4.1.1, 4.2.3, 5.0.2	
maxTcpConnsPerNodeConn	5.1.1		
metadataDiskWaitTimeForRecovery		4.1.0.4	
minDiskWaitTimeForRecovery	4.1.1		
minMissedPingTimeout	4.2.1		
mmapRangeLock			
mmfsLogTimeStampISO8601	4.2.2		
nfsPrefetchStrategy	4.2.1		
nistCompliance	4.1		
noSpaceEventInterval			
nsdBufSpace			
nsdCksumTraditional	5.0.1		
nsdDumpBuffersOnCksumError	5.0.1		
nsdInlineWriteMax	4.2.1		

Parameter¹	Added	Updated²	Obsoleted³
nsdMaxWorkerThreads	4.2.1		
nsdMinWorkerThreads	4.2.1		
nsdMultiQueue	4.2.1		
nsdRAIDTracks			
nsdRAIDBufferPoolSizePct			
nsdServerWaitTimeForMount			
nsdServerWaitTimeWindowOnMount			
numaMemoryInterleave	4.1.0.4	5.2.0	
pagepool		4.2, 5.2.0	
pagepoolMaxPhysMemPct			
panicOnIOHang	5.0.2		
pitWorkerThreadsPerNode	4.1.1		
prefetchPct	4.2.1		
prefetchPartitions	5.2.0		
prefetchThreads		4.1.0.4	
profile	4.1.1		
readReplicaPolicy	4.1.1	4.2.1	
release=LATEST		4.2.1	
restripeOnDiskFailure		4.2.1	
rpcPerfNumberDayIntervals	4.1		
rpcPerfNumberHourIntervals	4.1		
rpcPerfNumberMinuteIntervals	4.1		
rpcPerfNumberSecondIntervals	4.1		
rpcPerfRawExecBufferSize	4.1, 5.0.0		
rpcPerfRawStatBufferSize	4.1		
seqDiscardThreshold	4.2.1		
sharedTmpDir	5.0.1		
sidAutoMapRangeLength			
sidAutoMapRange			
subnets			
sudoUser	5.0.0		
syncBuffsPerIteration	4.2.1		
syncSambaMetadataOps	4.2.1		
systemLogLevel	4.1		
tiebreakerDisks		4.1, 4.2.3	

Parameter¹	Added	Updated²	Obsoleted³
tscCmdPortRange	5.0.0		
uidDomain			
unmountOnDiskFail		4.2	
verbsGPUDirectStorage	5.1.2		
verbsNumaAffinity	5.1.2		
usePersistentReserve			
verbsPorts		4.2.2, 5.0.0	
verbsHungRdmaTimeout	5.1.0		
verbsRdma			
verbsRdmaCm			
verbsRdmaPkey	4.2.3		
verbsRdmaRoCEToS	4.1.0.4		
verbsRdmaSend			
verbsRdmasesPerConnection		4.2, 4.2.1	5.0.0
verbsRdmasesPerNode		4.2.1	5.0.0
verbsRecvBufferCount	5.0.0		
verbsRecvBufferSize	5.0.0		
verbsSendBufferMemoryMB			5.0.0
workerThreads	4.2	4.2.2, 5.2.0	
worker1Threads		4.2, 5.2.0	
writebehindThreshold	4.2.1		

¹ For more information, see “[Changing the GPFS cluster configuration data](#)” on page 8.

² For more information about the updated parameters, see the *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

³ Parameter is not valid starting from the given release.

Chapter 6. Ensuring high availability of the GUI service

You need multiple GUI nodes to be configured in the system to ensure high availability of the GUI service. You also need to set up a cluster configure repository (CCR) when you plan to configure multiple GUI nodes in the cluster. The CCR is used to store certain important configuration details that must be shared among all GUI nodes.

The following figure illustrates the GUI high availability configuration with two GUI nodes.

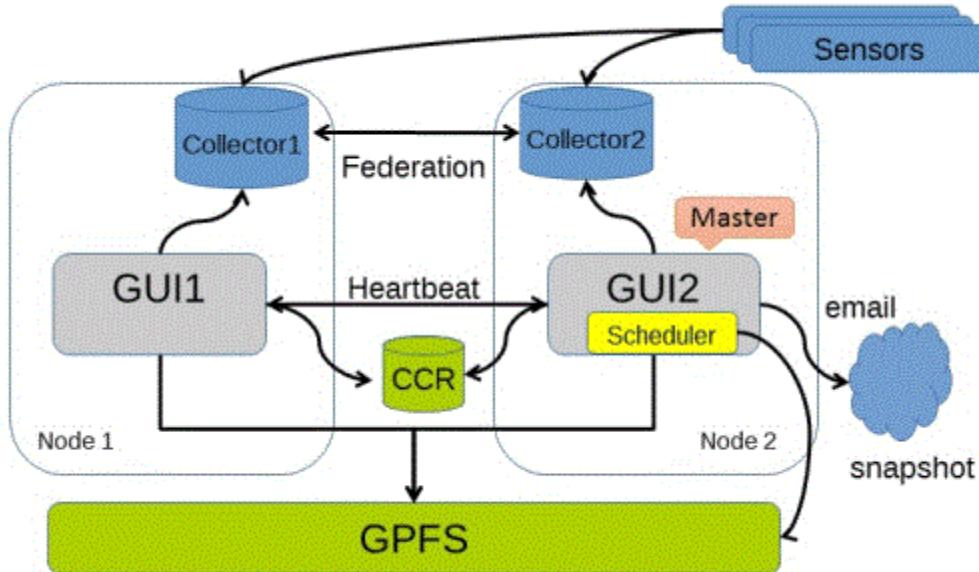


Figure 4. High availability configuration of GUI with two GUI nodes

The following list provides the configuration requirements to ensure high availability of the GUI service:

- Up to three GUI nodes can be configured in a cluster. Perform the following steps to set up a GUI node:
 - Install the GUI package on the node. For more information about latest packages, see *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
 - Start the GUI service and either log in or run `/usr/lpp/mmfs/gui/cli/initgui` to initialize the GUI database. Now, the GUI becomes fully functional and it adds the node to the `GUI_MGMT_SERVERS` node class.
- The GUI nodes are configured in the active/active configuration. All GUI nodes are fully functional and can be used in parallel.
- Each GUI has its own local configuration cache in PostgreSQL and collects configuration changes individually.
- One GUI node is elected as the master node. This GUI instance exclusively performs some tasks that must be run only once in a cluster such as running snapshot schedules, sending email, and SNMP notifications. If services that are run on the master GUI node are configured, the environment for all the GUI nodes must support these services on all nodes. For example, it needs to be ensured that access to SMTP and SNMP servers is possible from all GUI nodes and not only from the master GUI node. You can use the following utility function, which displays the current master GUI node:

```
[root@gpfsgui-11 ~]# /usr/lpp/mmfs/gui/cli/lsnode
Hostname          IP      Description        Role           Product  Connection GPFS   Last
updated
gpfsgui-11.novalocal 10.0.100.12 Master GUI Node management,storage 5.0.0.0  HEALTHY   HEALTHY 7/10/17
10:19 AM
```

gpfsgui-12.novalocal 10.0.100.13 10:19 AM	storage,ces	5.0.0.0	HEALTHY	HEALTHY	7/10/17
gpfsgui-13.novalocal 10.0.100.14 10:19 AM	storage,ces	5.0.0.0	HEALTHY	HEALTHY	7/10/17

- All GUI nodes are equal from the user's perspective. If a GUI node fails, the user must manually connect to the other GUI. The master role fails over automatically. But there is no failover for the IP address of the other GUI server.
- Data that cannot be gathered from GPFS is stored in CCR as shared-cluster repository. This includes GUI users, groups and roles, snapshot schedules, email notification settings, policy templates, and ACL templates.
- All GUI nodes must run on the same software level.
- If an external authentication method such as AD or LDAP is used to store the GUI user details and authenticate them, you must configure AD/LDAP on all GUI nodes to ensure high-availability. The Trustore key that is used during external authentication, must be created by using the public keys generated by all the GUI nodes in the cluster. If internal authentication method is used, the GUI nodes get the user information from the CCR.
- To display the performance monitoring information, install performance monitoring collector on each GUI node and these collectors must be configured in the federated mode. The data collection from the sensors can be configured in such a way that the details are sent either to all collectors or only to a single collector.
- The **Mark as Read** operation can be performed on events that are stored locally on the GUI node. The changes that are made to the events are not visible through the other GUI node.
- Each GUI has its own local configuration cache and collects configuration changes individually.
- A corrupted cache database affects only the local GUI. Other GUIs continue working. Most of the configuration changes are simultaneously reported in the GUI. Some configuration changes are gathered through the individually scheduled refresh tasks, which might result in displaying unsynchronized information.

Chapter 7. Configuring and tuning your system for Cloud services

This topic describes the procedure for configuring and tuning your IBM Storage Scale node for Cloud services.

Configuration command execution matrix

This topic describes which Cloud services commands can be run from which nodes. Typically, nodes can be categorized into 3 categories based on what software package (rpm) is installed on them. They are, Cloud services server node, Cloud services client node, and non-Cloud services node.

The following table provides the commands that can be run on various node categories:

Category	Command	Node Type		Non-Cloud services
		Cloud services server	Cloud services client	
Configuration	mmcloudgateway service start	y	Y	Y
	mmcloudgateway service status	y	Y	Y
	mmcloudgateway service stop	y	Y	Y
	mmcloudgateway service version	y	Y	Y
	mmcloudgateway service backupConfig	y		
	mmcloudgateway account *	Y	Y	Y
	mmcloudgateway cloudStorageAccessPoint *	Y	Y	Y
	mmcloudgateway clouddservice *	Y	Y	Y
	mmcloudgateway keymanager *	Y	Y	Y
	mmcloudgateway containerPairSet *	Y	Y	Y
	mmcloudgateway config *	Y	Y	Y
Data path	mmcloudgateway files migrate	Y	Y	
	mmcloudgateway files recall	Y	Y	
	mmcloudgateway files list	Y	Y	
	mmcloudgateway files restore	Y	Y	
	mmcloudgateway files delete	Y	Y	
	mmcloudgateway files destroy	Y	Y	
	mmcloudgateway files export	Y	Y	
	mmcloudgateway files import	Y	Y	

Category	Command	Node Type		Non-Cloud services
		Cloud services server	Cloud services client	
Maintenance	mmcloudgateway files cloudList	Y		
	mmcloudgateway files reconcile	Y	Y	
	mmcloudgateway files backupDB	Y		
	mmcloudgateway files checkDB	Y		
	mmcloudgateway files rebuildDB	Y	Y	
	mmcloudgateway files defragDB	Y		

Designating the Cloud services nodes

This topic describes how to designate a node as Cloud services node in the IBM Storage Scale cluster.

Before you begin, ensure that you install the server package RPMs on all nodes that you want to designate as Cloud services nodes. These nodes must have GPFS server licenses enabled.

Also, ensure that a user-defined node class is created and properly configured for Cloud services. For more information, see *Creating a user-defined node class for Transparent cloud tearing or Cloud data sharing in IBM Storage Scale: Concepts, Planning, and Installation Guide*.

To start working with Cloud services, the administrator first needs to designate a node as Cloud services node in the IBM Storage Scale cluster. Data migration to or data recall from a Cloud Object Storage that occurs in this node.

You can designate a maximum combination of four CES or NSD nodes as Cloud services nodes in each node class (with a maximum of four node class for 16 nodes total) in the IBM Storage Scale cluster.

If you use multiple node classes for Cloud services, then you can designate at least one node in each node class as Cloud services server nodes.

By default and by way of recommendation, Cloud services use the node IP addresses, not the CES IPs.

Note: You need to perform this procedure only on a single node where the server package is installed.

1. To designate the nodes as Cloud services nodes, issue a command according to this syntax: **mmchnode change-options -N {Node[,Node...] | NodeFile | NodeClass} [--cloud-gateway-nodeclass CloudGatewayNodeClass]**.

You can either choose to designate all nodes or only some selected nodes in a node class as Cloud services nodes.

To designate all nodes in the node class, TCTNodeClass1, as Cloud services server nodes, issue this command:

```
mmchnode --cloud-gateway-enable -N TCTNodeClass1
```

To designate only a few nodes (node1 and node2) in the node class, TCTNodeClass1, as Cloud services server nodes, issue this command:

```
mmchnode --cloud-gateway-enable -N node1,node2 --cloud-gateway-nodeclass TCTNodeClass1
```

It designates only node1 and node2 as Cloud services server nodes from the node class, TCTNodeClass1. Administrators can continue to use the node class for other purposes.

Note: The Cloud services node must have connectivity to the object storage service that the Cloud services uses.

2. To designate nodes from multiple node classes as Cloud services server nodes, issue the following commands:

- `mmchnode --cloud-gateway-enable -N TCTNodeClass1`
- `mmchnode --cloud-gateway-enable -N TCTNodeClass2`

Note: These nodes cannot be combined into a single Cloud services across node classes because Cloud services for nodes in different node classes are always different or separate.

3. To list the designated Transparent cloud tiering nodes, issue this command: **mmccloudgateway node list**.

Note: For more information, see the **mmccloudgateway** command in *IBM Storage Scale: Command and Programming Reference Guide*.

4. To disable two nodes, node1 and node2, from the node class, TCTNodeClass1, issue this command:

```
mmchnode --cloud-gateway-disable -N nod1,node2 --cloud-gateway-nodeclass TCTNodeClass1
```

You can add a node to the node class at any time. For example, issue the following commands to add the node, 10.11.12.13, to the node class, TCTNodeClass1:

1. **mmchnodeclass** TCTNodeClass1 add -N 10.11.12.13
2. **mmchnode** --cloud-gateway-enable -N 10.11.12.13 --cloud-gateway-nodeclass TCTNodeClass1

Starting up the Cloud services software

This topic describes how to start the Cloud services software on IBM Storage Scale nodes.

Before you try to start Cloud services on a node, ensure that the node is designated as a Cloud services node. For more information, see “[Designating the Cloud services nodes](#)” on page 88.

Start the Cloud services before you run any of the Cloud services commands.

To start Cloud services, issue a command according to this syntax:

```
mmccloudgateway service start [-N {alltct | Node[,Node...] | NodeFile | NodeClass}]
```

For example, to start the service on all Transparent cloud tiering nodes in a cluster, issue this command:

```
mmccloudgateway service start -N alltct
```

To start the service on all Cloud services nodes as provided in the node class, *TCTNodeClass1*, issue this command:

```
mmccloudgateway service start -N TCTNodeClass1
```

If you provide this command without any arguments, the service is started on the current node.

If you have more than one node class, then you must start the Cloud services individually on each node class, as follows:

- `mmccloudgateway service start -N TCTNodeClass1`
- `mmccloudgateway service start -N TCTNodeClass2`

It is a good practice to verify that the service is started. Enter a command like the following one:

```
mmccloudgateway service status -N TCTNodeClass
```

Note: You can run this command from any node in the cluster, not necessarily from a node that is part of a node class.

Next step: See “[Managing a cloud storage account](#)” on page 90.

Managing a cloud storage account

You can manage a cloud storage account by using the **mmcloudgateway account create/update/delete** command.

Note:

- Before you try to configure a cloud storage account, ensure that the Cloud services are started. For more information, see “[Starting up the Cloud services software](#)” on page 89.
- Before deleting a cloud storage account, ensure that you recall all the data that is migrated to the cloud.
- Ensure that Network Time Protocol (NTP) is enabled and time is correctly set.

Note: Even though you specify the credentials for the cloud account, the actual validation does not happen here. The authentication of the credentials happens only when you create a cloud storage access point. Therefore, you do not receive any authentication error even if you provide some wrong cloud account credentials.

Next step: See “[Defining cloud storage access points \(CSAP\)](#)” on page 92.

Amazon S3

Account creation for Amazon S3

Note:

- us-standard (us-east-1 N.Virginia)
- us-east-2 (Ohio)
- us-west-1 (N.California)
- us-west-2 (Oregon)
- eu-west-1 (Ireland)
- eu-west-2 (London)
- eu-west-3 (Paris)
- eu-central-1 (Frankfurt)
- eu-north-1 (Stockholm)
- sa-east-1 (Sao-Paulo)
- ap-southeast-1 (Singapore)
- ap-southeast-2 (Sydney)
- ap-south-1 (Mumbai)
- ap-northeast-1 (Tokyo)
- ap-northeast-2 (Seoul)
- ap-northeast-3 (Osaka)
- ca-central-1 (Canada)
- cn-north-1 (Beijing)
- cn-northwest-1 (Ningxia)

Do the following steps:

- To create a cloud account using Amazon S3, issue the following command:

```
mmcloudgateway account create --cloud-nodeclass tct --account-name s3account  
--cloud-type S3 --username AKIAISCW6DRRITWR6IWQ --pwd-file /tmp/cloudPW
```

The system displays the following output:

```
mmcloudgateway: Sending the command to the first successful node starting with  
jupiter-vm716.pok.stglabs.ibm.com  
mmcloudgateway: This may take a while...
```

```
mmcloudgateway: Command completed successfully on jupiter-vm716.pok.stglabs.ibm.com.  
mmcloudgateway: You can now delete the password file '/tmp/cloudPW'  
mmcloudgateway: Command completed
```

Note: For Amazon S3, the `--username` represents the access key.

- To modify the cloud account (for example, to change the username to `MyTct`) issue the following command:

```
mmcloudgateway account update --cloud-nodeclass tct --account-name s3account  
--username MyTct
```

- To delete a cloud account, issue the following command:

```
mmcloudgateway account delete --cloud-nodeclass tct --account-name s3account
```

Swift3 account

Account creation for Swift3

Do the following steps:

- To configure a cloud storage tier for Swift3, issue this command:

```
mmcloudgateway account create --cloud-nodeclass tct --account-name Swift3account  
--cloud-type SWIFT3 --username 92d32006d1214eee9f97eb47ffdf8f6d --pwd-file /tmp/cloudPW
```

The system displays the following output:

```
mmcloudgateway: Sending the command to the first successful node starting with  
ip9-114-192-175.pok.stglabs.ibm.com  
mmcloudgateway: This may take a while...  
mmcloudgateway: Command completed successfully on ip9-114-192-175.pok.stglabs.ibm.com.  
mmcloudgateway: You can now delete the password file '/tmp/cloudPW'  
mmcloudgateway: Command completed.
```

- To modify the account details (for example, to modify the user name), issue this command:

```
mmcloudgateway account update --cloud-nodeclass tct --account-name Swift3account  
--username Testuser1
```

IBM Cloud Object Storage

Managing account creation for IBM Cloud® Object Storage

Note: While using nginx as a load balancer with IBM Cloud Object Storage, ensure that invalid-headers and etag attributes are turned off for Transparent cloud tiering to work correctly. Without these settings, any Transparent cloud tiering request to IBM Cloud Object Storage would fail with errors that indicate signature mismatch.

Do the following steps:

- To configure a new cloud object storage account for the IBM Cloud Object Storage version 3.7.2 and above, enter a command like the following:

```
mmcloudgateway account create --cloud-nodeclass TCTNodeClass1 --account-name  
cscloud --cloud-type CLEVERSAFE-NEW --username user1 --pwd-file MyFile.txt
```

Note: The username represents the access key.

The system displays output similar to this:

```
mmcloudgateway: Sending the command to the first successful node starting with  
vmip51.gpfs.net  
mmcloudgateway: This may take a while...  
mmcloudgateway: Command completed successfully on vm1.gpfs.net.
```

```
mmcloudgateway: You can now delete the password file 'MyFile.txt'  
mmcloudgateway: Command completed.
```

- To create a cloud account for deploying a WORM solution by using locked vaults, issue a command like the following:

```
mmcloudgateway account create --cloud-nodeclass NodeClass1 --account-name myCloud  
--cloud-type CLEVERSAFE-NEW  
--src-keystore-path /root/test/testalias.ssl/testalias.jks  
--src-alias-name testalias --src-keystore-type JKS  
--src-keystore-pwd-file /root/pwd/file.txt
```

- To update an account:

```
/usr/lpp/mmfs/bin/mmcloudgateway account update --cloud-nodeclass tct  
--account-name cleversafeaccount --username 5n0YjGWDvNiQP0ZsmzGl  
--pwd-file /tmp/cloudPW
```

- To delete an account:

```
mmcloudgateway account delete --cloud-nodeclass tct --account-name  
cleversafeaccount
```

Microsoft Azure

This section informs you how to create account for Microsoft Azure.

For Azure, cloud services tier or share data only to 'cool' storage tier by default. Other Azure tiers such as hot, premium, and archive are not supported. Customization is not supported.

Similarly, cloud services tier or share data only as a 'block' blob by default. Other blob types such as append and page blobs are not supported. Customization is not supported.

Complete the following steps:

To create an Azure cloud account, issue the following command:

```
mmcloudgateway account create --cloud-nodeclass tct --account-name azureaccount --cloud-type  
azure --username <azure-storage-account-name> --pwd-file <passwd-file>
```

The system displays the following output:

```
mmcloudgateway: Sending the command to the first successful node starting with  
jupiter-vm716.pok.stglabs.ibm.com  
mmcloudgateway: This may take a while...  
mmcloudgateway: Command completed successfully on jupiter-vm716.pok.stglabs.ibm.com.  
mmcloudgateway: You can now delete the password file '/tmp/cloudPW'  
mmcloudgateway: Command completed
```

Note: For Azure, the --username represents the storage account name.

To modify the cloud account (for example, to change the username to MyTct) issue the following command:

```
mmcloudgateway account update --cloud-nodeclass tct --account-name azureaccount --username  
MyTct
```

To delete a cloud account, issue the following command:

```
mmcloudgateway account delete --cloud-nodeclass tct --account-name azureaccount
```

Defining cloud storage access points (CSAP)

The Cloud Storage Access Point (CSAP) provides access between the cloud account on your object storage and IBM Storage Scale. You must create at least one CSAP per cloud account so your Cloud services have a path to the object storage. Extra CSAPs can also be created. CSAPs that all have about

the same lowest latency (which is tested every 30 minutes) to a node are used evenly. CSAPs with higher latency are put in standby and are used only in error scenarios. This provides greater throughput and higher availability in various error scenarios.

You can send data to the cloud object storage through Cloud Storage Access Points (CSAPs). Each cloud account needs at least one CSAP defined to have a path to the cloud. For some cases (IBM SoftLayer®, Amazon S3, or cloud storage with a load balancer with built-in redundancy) one accessor suffices. However, for cases where traffic is going directly to the object storage, it is usually beneficial to have more than one CSAP to provide needed availability and bandwidth for performance. For example, if you are designing an on-premises solution with IBM Cloud Object Storage, you would need to create one access point for each accessor node you want to send data to. The Cloud services would randomly assign work to the available accessors as long as they perform properly (broken or slow access points are avoided).

Note: If multiple intermediate certificates are issued by an internal certifying authority (CA), ensure to provide only a self-signed internal CA rather than providing a file that contains all the intermediate certificates. For example, if the CA issued a certificate chain such as Internal CA->cert1->cert2, then the input pem file must contain only the Internal CA certificate.

To create, update, or delete a CSAP:

- To create a CSAP according to the cloud account that is created, issue a command similar to this:

```
mmcloudgateway cloudStorageAccessPoint create --cloud-nodeclass TCTNodeClass1  
--cloud-storage-access-point-name AccessPoint1  
--account-name mycloud --url http://192.0.2.0
```

The system displays output similar to this:

```
mmcloudgateway: Sending the command to the first successful node starting with vmip51.gpfs.net  
mmcloudgateway: This may take a while...  
mmcloudgateway: Command completed successfully on vmi.gpfs.net.  
mmcloudgateway: Command completed.
```

- To create a CSAP with an https endpoint, issue a command similar to this:

```
mmcloudgateway cloudStorageAccessPoint create --cloud-nodeclass TCTNodeClass1  
--cloud-storage-access-point-name AccessPoint1  
--account-name mycloud --url https://192.0.2.0 --server-cert-path /root/ca.pem
```

- To delete a CSAP, issue a command similar to this:

```
mmcloudgateway cloudStorageAccessPoint delete --cloud-nodeclass cloud  
--cloud-storage-access-point-name csap1
```

The system displays output similar to this:

```
mmcloudgateway: Sending the command to the first successful node starting with vmip51.gpfs.net  
mmcloudgateway: This may take a while...  
mmcloudgateway: Command completed successfully on vmip51.gpfs.net.  
mmcloudgateway: Command completed.
```

Note:

- In proxy-based environments, set your proxy settings as part of the node class configuration before you run any migrations. If tiering commands (migrate or recall) are run before you set the proxy details, they might fail for not being able to reach out to the public cloud storage providers such as Amazon S3.
- To work with a specific region, specify the `--region` parameter while the new CSAP is created. Provide the same value with `--data-location` and `--meta-location` parameters, when you create container pair set later.

For example, if the administrator specifically wants to use ap-south-1 AWS S3 region, use ap-south-1 value for `--region` parameter while you create the CSAP. Also, specify `--data-location ap-south-1` and `--meta-location ap-south-1` parameters, when you create container pair set later.

If you fail to specify the parameters during container pair set creation, you might end up creating the container pair set in the default region (us-east-1). If the default region is not available, the container pair set creation would fail with the **No online CSAP found** error.

For more information, see the *mmcloudgateway command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Next step: See “[Creating Cloud services](#)” on page 94.

Creating Cloud services

You must create at least one Cloud services for each cloud account that is created. You can associate a single Cloud services with only one cloud account but can use it with multiple file systems or file sets.

For data movement (sharing or tiering) commands, you must specify a Cloud service name to move data to the intended object storage if there is more than one Cloud service that is configured to the file system or file set. However, you do not have to specify a Cloud service name for these data movement commands if only one Cloud service is configured for a file system or file set.

Additionally, if you want to execute both tiering and sharing operations in your Cloud services setup, you must define one Cloud service for tiering and another for sharing.

- To create a Cloud service according to the cloud account that is created, issue a command similar to the following one:

```
mmcloudgateway cloudService create  
    --cloud-nodeclass TCTNodeClass1 --cloud-service-name mycloud  
    --cloud-service-type Tiering --account-name Cleversafe_cloud
```

The system displays output similar to this:

```
mmcloudgateway: Sending the command to the first successful node starting with vmip51(gpfs.net  
mmcloudgateway: This may take a while...  
mmcloudgateway: Command completed successfully on vmi(gpfs.net.  
mmcloudgateway: Command completed.
```

Note: You can use this Cloud service only for tiering. If you want to use it for sharing, you can replace *Tiering* with *Sharing*.

- To update Cloud services, issue a command according to the following:

```
mmcloudgateway cloudService update --cloud-nodeclass cloud --cloud-service-name newServ --  
    disable
```

The system displays output similar to this:

```
mmcloudgateway: Sending the command to the first successful node starting with vmip51(gpfs.net  
mmcloudgateway: This may take a while...  
mmcloudgateway: Command completed successfully on vmip51(gpfs.net.  
mmcloudgateway: Command completed.
```

- To delete Cloud services, issue a command according to the following:

```
mmcloudgateway cloudService delete --cloud-nodeclass cloud --cloud-service-name newServ
```

The system displays output similar to this:

```
mmcloudgateway: Sending the command to the first successful node starting with vmip51(gpfs.net  
mmcloudgateway: This may take a while...  
mmcloudgateway: Command completed successfully on vmip51(gpfs.net.  
mmcloudgateway: Command completed.
```

Next step: [“Configuring Cloud services with SKLM \(optional\)” on page 95.](#)

Configuring Cloud services with SKLM (optional)

To encrypt data that is tiered to the cloud storage, you need to first configure a key manager, before creating a container pair set in the next step. Two types of key manager are supported - local key manager (simple JCEKS based one) and IBM Security Key Lifecycle Manager (SKLM) server. You need to create one local key manager per cluster. The SKLM key manager is optional and might be created per cluster or per sets of file systems, depending on your security needs.

Before you configure Cloud services with IBM Security Key Lifecycle Manager, ensure that an SKLM server is installed. For more information, see [“Preparation for encryption” on page 742](#).

Note:

- Transparent cloud tiering supports only IBM Security Key Lifecycle Manager versions 2.6.0 and 2.7.0.
- Transparent cloud tiering cannot communicate with IBM Security Key Lifecycle Manager server that does not support TLSv1.2.

You can create a key manager when you want to use this parameter while you configure a container pair set in the next topic.

- To create an SKLM key manager, issue a command similar to the following command:

```
mmcloudgateway keymanager create --cloud-nodeclass cloud  
    --key-manager-name vm1  
    --key-manager-type RKM  
    --sklm-hostname vm1  
    --sklm-port 9080  
    --sklm-adminuser SKLMAAdmin  
    --sklm-groupname tct
```

The system displays output similar to the following:

```
Please enter a password:  
Confirm your password:  
mmcloudgateway: Sending the command to the first successful node starting with vmip51.gpfs.net  
mmcloudgateway: This may take a while...  
mmcloudgateway: Command completed successfully on vmip51.gpfs.net.  
mmcloudgateway: Command completed.
```

- To rotate a key manager, issue a command according to the following:

```
mmcloudgateway keymanager rotate --cloud-nodeclass cloud --key-manager-name vmip131
```

The system displays output similar to the following:

```
mmcloudgateway: Sending the command to the first successful node starting with c01.gpfs.net  
mmcloudgateway: This may take a while...  
mmcloudgateway: Command completed successfully on c80f4m5n01.gpfs.net.  
mmcloudgateway: Command completed.
```

- To update a key manager, issue a command according to the following:

```
mmcloudgateway keymanager update --cloud-nodeclass cloud  
    --key-manager-name sklm --update-certificate
```

The system displays output similar to this:

```
mmcloudgateway: Sending the command to the first successful node starting with c01.gpfs.net  
mmcloudgateway: This may take a while...  
mmcloudgateway: Command completed successfully on c01.gpfs.net.  
mmcloudgateway: Command completed.
```

Next step: [“Binding your file system or fileset to the Cloud service by creating a container pair set” on page 96](#).

Note: The local key manager is simpler to configure and use. It might be your best option unless you are already using SKLM in your IBM Storage Scale cluster or in cases where you have special security requirements that require SKLM.

Binding your file system or fileset to the Cloud service by creating a container pair set

Up to this point, the configuration work was about creating access out to the cloud. Now, you need to bind all this to the data on your cluster, and you do that by creating a container pair set that associates your file system or fileset to the Cloud service. Once you create your container pair set, your Cloud service is usable.

Cloud services internally creates two containers on cloud storage for storing data as well as meta-data. However, some cloud providers require containers to be created using its native interfaces. In that case, you need to provide the names. The containers that are created for Cloud data sharing can be shared with other file systems or Cloud services. However, the containers that are created for tiering cannot be shared. Creating the container pair set is how you bind a file system to a Cloud service. Note that all file sets being bound to a file system must be assigned to the same Cloud services node class.

Note: When the existing tiering container reaches default 100,000,000 (100 million) entries or configured non-default threshold, and if auto-spillover is enabled, the Cloud service automatically creates a new container during the next reconcile operation. As an administrator, you can still create a new container for the same path, even before reaching the spillover threshold. After a new container is created for the configured filesystem or fileset path or auto-spillover, the previous container goes to the Inactive state and new migrations go to the newly created container. Creation of a new container only affects target container for new migrations. Recalls are unaffected and continue from the container (including inactive containers) where data was migrated. If auto-spillover (new container creation) fails for any reason, the current active container indicates overdue status for container spillover.

Containers by default do not have encryption enabled. If encryption at rest is required, the best performance will be there by enabling encryption natively at the object storage and not having Cloud service encrypt the data which is what Cloud service does if encryption is enabled in the commands below. Note that encryption on the wire is covered by Cloud service even if encryption is disabled because Cloud service uses https to send the data.

Note: Administrators need to explicitly add "--enc ENABLE" parameter while creating a container pair set, to ensure that the data is encrypted while being tiered to a cloud storage.

If you have applications that frequently access the front end of the file, you might want to consider enabling thumbnail support. An example of an application that accesses the first few bytes of a file is Windows Explorer in order to provide a thumbnail view of image files. There is no limit on the amount of data you can cache as the appropriate cache size is application specific. You can create a container pair set with thumbnail enabled, and the scope can be enabled to either a file system or a fileset according to your business requirements.

Note:

- Changing the mount point for a file system or the junction path of a fileset after it is associated with a container is not supported.
- You can enable or disable transparent recall policy by using the --transparent-recalls {ENABLE|DISABLE} parameter. However, this parameter is optional, and transparent recall policy is enabled by default even if you do not use this parameter.

Do the following steps for creating, testing, listing, or deleting a container pair set:

- To create a container pair set, issue a command similar to this:

```
mmcloudgateway containerpairset create --cloud-nodeclass TCTNodeClass1  
--container-pair-set-name newContainer  
--cloud-service-name myService  
--scope-to-filesystem  
--path Path
```

```
--data-container DataContainer  
--meta-container MetaContainer
```

The system displays output similar to this:

```
mmcloudgateway: Sending the command to the first successful node starting with v1.gpfs.net  
mmcloudgateway: This may take a while...  
mmcloudgateway: Command completed successfully on vmi.gpfs.net.  
mmcloudgateway: Command completed.
```

Note: If you do not specify the names for storing data and metadata containers, then the container pairset name is used for both data and meta containers. In this example, they are "newContainer" (for data) and "newContainer.meta" (for metadata). If you create any meta-containers or data-containers by using any external tools, you can configure containerpairset with these meta-containers or data-containers. To know the bucket creation rules while naming meta-container and data-container for ICOS, S3 and AWS S3 provider, see [Rules for Bucket Naming at Bucket restrictions and Limitations](#).

- To create a container pairset with thumbnail enabled and the scope is a file system, issue a command similar to this:

```
mmcloudgateway containerpairset create --cloud-nodeclass cloud --container-pair-set-name x13  
--cloud-service-name newServ --scope-to-filesystem --path /gpfs --thumbnail-size 64
```

- To create a container pairset when the scope is a fileset, issue a command similar to this:

```
mmcloudgateway containerpairset create --cloud-nodeclass cloud --container-pair-set-name x13  
--cloud-service-name newServ --scope-to-fileset --path /gpfs/myfileset
```

- To create a container pair set that is enabled for encryption, issue a command similar to this:

```
mmcloudgateway containerpairset create --cloud-nodeclass tct --container-pair-set-name Containeretag5  
--cloud-service-name csss5 --path /gpfs0/fs3 --enc ENABLE --etag ENABLE  
--data-container test5 --meta-container testmeta5 --key-manager-name lkm3 --scope-to-fileset  
--path /gpfs/myfileset
```

- To configure a container pair set using an immutable fileset with a fileset scope, issue a command similar to this:

```
mmcloudgateway containerpairset create --cloud-nodeclass tct --container-pair-set-name wormcp  
--cloud-service-name wormservice2 --path /gpfs0/worm2 --enc ENABLE --etag ENABLE --data-  
container wormtestnov --meta-container wormtestnovmeta --key-manager-name lkm3 --scope-to-fileset  
--path /gpfs/myfileset --cloud-directory-path /gpfs0/fs3
```

Note: Here, the fileset is an immutable fileset whereas the cloud directory is pointing to a fileset that is not immutable.

- To test a container pair set that is created, issue a command similar to this:

```
mmcloudgateway containerpairset test --cloud-nodeclass cloud --container-pair-set-name vmip51
```

Note: This test will check whether or not the container pair set does actually exist. Additionally, the test will try to add some data to the container (PUT blob), retrieve the data (GET blob), delete the data (DELETE blob), and report the status of each of these operations. This test will validate whether or not all CSAPs for a given container pair set are able to reach the cloud storage.

- To delete a container pair set, issue a command similar to this:

```
mmcloudgateway containerpairset delete --container-pair-set-name x13  
--cloud-nodeclass cloud
```

- To list a container pair set, issue a command similar to this:

```
mmcloudgateway containerpairset list
```

The system displays output similar to this:

```
Configured 'containerPairSet' options from node class  cloud:
-----
  containerPairSetName      : vmip51
  path                      : /gpfs/
  scopeTo                  : filesystem
  transparentRecalls       : Enabled
  cloudServiceName         : newServ
  dataContainer             : vmip51
  metaContainer             : vmip51.meta
  thumbnailSize             : 0
  cloudDirectoryPath        : /fs/
  dataLocation               :
  metaLocation               :
  activeKey                 :
  keyManagerName            :
  containerPairSetName       : fset1
  path                      : /gpfs/
  scopeTo                  : filesystem
  transparentRecalls       : Disabled
  cloudServiceName         : newServ
  dataContainer             : fset1
  metaContainer             : fset1.meta
  thumbnailSize             : 0
  cloudDirectoryPath        : /fs/
  dataLocation               :
  metaLocation               :
  activeKey                 :
  keyManagerName            :
  policyTempDir             : /gpfs1
Configured 'containerPairSet' options from node class cloud2:
-----
  containerPairSetName      : vmip53
  path                      : /x13/
  scopeTo                  : filesystem
  transparentRecalls       : Enabled
  cloudServiceName         : serv2
  dataContainer             : vmip53
  metaContainer             : vmip53.meta
  thumbnailSize             : 0
  cloudDirectoryPath        : /x13/
  dataLocation               :
  metaLocation               :
  activeKey                 :
  keyManagerName            :
  policyTempDir             : /gpfs1
```

- To create a container pair set with auto-spillover disabled, issue a command similar to this:

```
mmcloudgateway containerpairset create --cloud-nodeclass tct --container-pair-set-name
spilloverdisabled --cloud-service-name csss5 --path /gpfs0/fs3
--scope-to-fileset --auto-spillover DISABLE
```

- To create a container pair set with a non-default threshold value for auto-spillover, issue a command similar to this:

```
mmcloudgateway containerpairset create --cloud-nodeclass tct
--container-pair-set-name spilloverdisabled
--cloud-service-name csss5 --path /gpfs0/fs3
--scope-to-fileset --auto-spillover-threshold <non-default-value>
```

To avoid too many spillovers, lower value for threshold is 50 million, whereas upper limit is 100 million entries per container. Non-default value entered in the preceding example, must be within this range. Run **mmcloudgateway containerpairset list** to list new container spillover attributes.

Note: Now that you have created your container configuration, it is critical that you do the following:

- Back up your configuration and security information for disaster recovery purposes. For more information, see [“Scale out backup and restore \(SOBAR\) for Cloud services” on page 889](#).

- A review of background container pair set maintenance activities is highly recommended. For more information, see the *Planning for maintenance activities* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

For more information, see the *mmcloudgateway command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Next step: [“Backing up the Cloud services configuration” on page 99](#)

Backing up the Cloud services database to the cloud

Transparent cloud tiering database stores critical information, such as the list of files that are migrated to the cloud and the list of files that are deleted from the cloud, which is associated with each container. It is essential to back up this database for any type of disaster recovery.

To back up the Transparent cloud tiering database, issue a command according to the following syntax:

```
mmcloudgateway files backupDB --container-pair-set-name <coontainerpairsetname>
```

For example, to back up the database that is associated with the container, `cpair1`, issue this command:

```
mmcloudgateway files backupDB --container-pair-set-name cpair1
```

The system displays output similar to this:

```
mmcloudgateway: Command completed.
```

If the database size is large, then backing up operation can be a long running process.

Note: By using the backed-up database, you can perform a database recovery by using the **mmcloudgateway files rebuildDB** command. For more information, see [“Manual recovery of Transparent cloud tiering database” on page 888](#).

Backing up the Cloud services configuration

It is critical that the Cloud services configuration data that is stored in the CCR is always backed up.

To back up the configuration data, issue a command according to the following syntax:

```
mmcloudgateway service backupConfig --backup-file <name of the file including the path>
```

The following files are backed up from the CCR:

- `mmcloudgateway.conf`
- `_tctkeystore.jceks`
- `_nodegroup1.settings`
- `_nodegroup2.settings`

Note: Files that are collected as part of backup are settings file for each node group.

You can specify a path along with the file name. If the path does not exist, then the command creates the path. The backed-up files are stored in a tar file, which is saved under the specified folder. If the path is not specified, then the tar file is stored in the local directory.

Refer to the following examples:

- Issue the following command to back up the configuration data to the file, `tct_config_backup` under `/tmp/mydir/`, where the folder does not exist:

```
mmcloudgateway service backupConfig --backup-file /tmp/mydir/tct_config_backup
```

The system displays output similar to this:

```
mmcloudgateway: Directory '/tmp/mydir' does not exist. Creating it.  
mmcloudgateway: Starting backup  
Backup Config Files:  
[mmcloudgateway.conf - Retrieved]  
[_tctkeystore.jceks - Not Found]  
[_cloudnodeclass.settings - Retrieved]  
[_cloudnodeclass1.settings - Retrieved]  
  
mmcloudgateway: Creating the backup tar file...  
mmcloudgateway: Backup tar file complete. The file is '/tmp/mydir/  
tct_config_backup_20170915_085741.tar'  
mmcloudgateway: The backup file should be archived in a safe location.  
mmcloudgateway: Command completed.
```

- Issue the following command to back up the configuration data to the file, `tct_config_backup`, under `/tmp/mydir/`, where the folder does exist:

```
mmcloudgateway service backupconfig --backup-file /tmp/mydir/tct_config_backup
```

The system displays output similar to this:

```
mmcloudgateway: Starting backup  
Backup Config Files:  
[mmcloudgateway.conf - Retrieved]  
[_tctkeystore.jceks - Not Found]  
[_cloudnodeclass.settings - Retrieved]  
[_cloudnodeclass1.settings - Retrieved]  
  
mmcloudgateway: Creating the backup tar file...  
mmcloudgateway: Backup tar file complete. The file is '/tmp/mydir/  
tct_config_backup_20170915_085741.tar'  
mmcloudgateway: The backup file should be archived in a safe location.  
mmcloudgateway: Command completed.
```

In these examples, `tct_config_backup` is the name that is given to the tar file. The file name is appended with the date and time when the command is run. The format is `filename_yyyymmdd_hhmmss.tar`. By doing so, the file names are not overwritten even if an administrator runs this command multiple times, providing the same file name.

Note: It is a best practice to save the backup file in a safe location outside the cluster to ensure that the backup file can be retrieved even if the cluster goes down. For example, when you use encryption no copy of the key library is made to cloud storage by Transparent cloud tiering. Therefore, if there is a disaster in which a cluster is destroyed, you must make sure that the key library is safely stored on a remote cluster. Otherwise, you cannot restore the Transparent cloud tiering service for files that are encrypted on the cloud because the key to decrypt the data in the cloud is no longer available.

A good way to back up the Cloud services configuration is as a part of the SOBAR based backup and restore script that is included. For more information, see [“Scale out backup and restore \(SOBAR\) for Cloud services” on page 889](#).

Configuring the maintenance windows

This topic describes the procedure for setting up a maintenance window in Cloud services.

The regular maintenance tasks are backing up the cloud database, reconciling files between the file system and the object storage, and deleting cloud objects that are marked for deletion. These activities are automatically done by Cloud services according to the default schedules. However, if the default schedules do not suit your requirements, you can modify the schedules and create your own maintenance windows by using the `mmcloudgateway maintenance` command.

Before setting up a maintenance window, review the guidelines provided here: *Planning for maintenance activities* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

To configure a maintenance window, do the following steps:

- To view the maintenance status, run a command as follows:

```
mmcloudgateway maintenance status --cloud-nodeclass tct
```

The system displays the following output:

```
=====
Maintenance status from node class tct:
=====

Summary:
=====
Total Containers : 1
Total Overdue : 0
Total In Progress : 0
Reconcile Overdue : 0
Backup Overdue : 0
Retention Overdue : 0
=====

Container: producer-container
=====
Status : Active
In Progress : no
File Count : 5
Files Deleted (last run) : 0

Maintenance Details: Reconcile Backup Retention
-----
Status : OK OK OK
Last Attempted : 01:00 04/03/2018 01:00 04/03/2018 01:00 04/10/2018
Last Successful : 01:00 04/03/2018 01:00 04/03/2018 01:00 04/10/2018
Time Ran (mins) : 1 1 -
```

- To create a daily maintenance windows, run a command as follows:

```
mmcloudgateway maintenance create --cloud-nodeclass cloud --maintenance-name main1
--daily 12:00-13:00
```

The system displays output similar to this:

```
mmcloudgateway: Sending the command to the first successful node starting with
c80f4m5n01.gpfs.net
mmcloudgateway: This may take a while...
mmcloudgateway: Command completed successfully on c80f4m5n01.gpfs.net.
mmcloudgateway: Command completed
```

- To create a weekly maintenance window, run a command as follows:

```
mmcloudgateway maintenance create --cloud-nodeclass cloud --maintenance-name main2
--weekly 1:07:00-2:07:00
```

The system displays output similar to this:

```
mmcloudgateway: Sending the command to the first successful node starting with
c80f4m5n01.gpfs.net
mmcloudgateway: This may take a while...
mmcloudgateway: Command completed successfully on c80f4m5n01.gpfs.net.
mmcloudgateway: Command completed.
```

- To list the configured maintenance schedule, run a command as follows:

```
mmcloudgateway maintenance list
```

The system displays output similar to this:

```
Configured maintenance options from node class cloud:
-----
maintenanceName      : defaultDaily
type                : daily
startTime           : 01:00
```

```

endTime : 04:00
enabled : true

maintenanceName : defaultWeekly
type : weekly
startTime : 6:01:00
endTime : 1:01:00
enabled : true

maintenanceName : main1
type : daily
startTime : 08:00
endTime : 09:00
enabled : true

maintenanceName : main2
type : weekly
startTime : 1:07:00
endTime : 2:07:00
enabled : true

taskFrequencyName : default
backupFrequency : weekly
reconcileFrequency : monthly
deleteFrequency : daily

```

- To update the maintenance schedule, issue a command as follows:

```
mmcloudgateway maintenance update --cloud-nodeclass cloud --maintenance-name dd --daily 08:00-09:00
```

- To delete a maintenance schedule, issue a command as follows:

```
mmcloudgateway maintenance delete --cloud-nodeclass 99 --maintenance-name main1
```

- To disable the maintenance schedule, issue a command as follows:

```
mmcloudgateway maintenance setState --cloud-nodeclass cloud --maintenance-name main2 --state disable
```

When you check the output of the **mmcloudgateway maintenance list** command, you can see the status of the **enabled** field as *false*.

Note: Disabling maintenance activities permanently is not recommended nor is it a supported mode of operation.

- To set the frequency of a specific maintenance operation, issue a command as follows:

```
mmcloudgateway maintenance setFrequency --cloud-nodeclass cloud --task reconcile --frequency daily
```

By default, all operations (reconcile, backup, and delete) are done according to the default frequency when a maintenance task is run. You can use the *setFrequency* option to modify the default frequency of a specific operation. For example, the default frequency for the reconcile operation is monthly, but you can change the frequency of the reconcile operation to weekly. The default frequency for the backup operation is weekly. After every couple of million files, a backup operation must be performed. If you observe heavy backups with the default frequency, perform manual backups or set the backup frequency to daily.

When a daily, weekly, or monthly frequency is specified for an operation, what it really means is that the operation will be executed no more often than its specified frequency. So, for example, an operation with a daily frequency will run no more often than once per day.

Enabling a policy for Cloud data sharing export service

This topic describes how to create a policy for export, a service provided by the IBM Storage Scale Cloud services, Cloud data sharing.

After you create a cloud storage account, you will want to create a policy for exporting data to cloud storage. A sample policy is provided below. You can run this policy manually as needed or set it up to run

periodically – a cron job is frequently employed for this purpose. This policy is meant to run weekly and exports files greater than one day old and less than eight days old.

Note: Do not set this down to modified age of 0 if you want to run it in a cron job -- as it will only pick up a partial list of files for day 0 and you may end up with gaps or duplicates over your week-to-week policy runs.

```
define(  
modified_age,  
(DAYS(CURRENT_TIMESTAMP) - DAYS(MODIFICATION_TIME))  
)  
RULE EXTERNAL LIST 'export' EXEC '/opt/ibm/MCStore/bin/mcstore' OPTS '-e'  
RULE 'files-to-export'  
LIST 'export'  
WHERE modified_age > 1 AND modified_age <= 8
```

How this could be applied using a cron job:

1. Open a cron job editor by issuing this command:

```
crontab -e
```

2. Add weekly export cron job by specifying this command:

```
@weekly mmapplypolicy {Device|Directory} -P PolicyFile
```

3. Specify a command to export the journal to cloud storage here, if required.
4. Specify the asynchronous notification of new files in the cloud here (pick your favorite notification tool).
5. Save the file.

Tuning Cloud services parameters

This topic describes the tunable parameters for Cloud services and the commands to modify them.

Cloud services use some default configuration parameters. You can change the value of the parameters if the default settings do not suit your requirements.

The following table provides the list of configurable parameters and their description.

Table 5. Attributes and default values				
Variable Name	Default Value	Minimum Value	Maximum Value	Description
connector.server.timeout	5000 (ms)	1000 (ms)	15000 (ms)	This is the maximum amount of time the server takes to respond to the client request. If the request is not fulfilled, it closes the connection.
connector.server.backlog	0	0	100	The maximum queue length for incoming connection indications (a request to connect) is set to the backlog parameter. If a connection indication arrives when the queue is full, the connection is refused.
destroy.sql.batchsize	8196	8196	81960	Page size per delete local database objects operation.
destroy.cloud.batchsize	256	256	81960	Page size per cloud objects delete operation.
reconcile.sql.batchsize	8196	8196	81960	Reconcile processes files in batches. This parameter controls how many files are processed in a batch.
commands.reconcile.lockwait.timeout.sec	360(s)	60(s)	3600(s)	Maximum time to acquire the lock on directory for the reconcile operation.

Table 5. Attributes and default values (continued)

Variable Name	Default Value	Minimum Value	Maximum Value	Description
threads.gc.batchsize	4096	4096	40960	Page size of the Garbage Collector thread. We can increase this in case the memory usage is more.
migration.downgrade.lock.threshold.size.mb	64 (MB)	1 (MB)	64 (MB)	Sets the size threshold on files for which the lock downgrade is completed. To save time, a lock downgrade is not completed on shorter files that can transfer quickly. For larger files, a lock downgrade is suggested because migration might take a long time.
cloud-retention-period-days	30	0	2147483647	Number of days for which the migrated data needs to be retained on the cloud after its file system object has been deleted or reversioned.
connector.server.migrate-threadpool-size	32	1	64	Thread pool size of the migration threads. User can do this by making sure the CPU resources (CPU speed and number of cores) of the Cloud service nodes match.
connector.server.recall-threadpool-size	32	1	64	Thread pool size of recall threads. User can increase the number of recall threads to improve the performance.
tracing.enable	true	true	False	Enables administrators to print trace messages of the internal components in a file. Controls the level of messages such as Info, Warning, or Error.
tracing.level	ALL=4	See Table 6 on page 104	See Table 6 on page 104	Tracing level is to set non-default tracing levels for various Transparent Cloud Tiering internal components to generate more debug data if any problems occur.
audit.enable	true	true	true	Enables or disables auditing information.
threads.cut-slow.sleep.ms	6000000 (ms)	60000 (ms)	2^63	Sleep time between two slow cloud update thread runs.
threads.cut-slow.sizediff	268435456 (Bytes)	1048576 (Bytes)	2^63	Size threshold of Cloud Updater Slow Threads.
threads.cut-slow.timediff.ms	604800000 (ms)	86400000 (ms)	2^63	Time threshold of Cloud Updater Slow Threads to update the cloud database.
threads.cut-fast.sizediff	16777216 (Bytes)	1048576 (Bytes)	2^63	Size threshold of Cloud Updater Fast Threads.
threads.cut-fast.timediff-active.ms	1800000 (ms)	60000 (ms)	2^63	Active time of Cloud Updater Fast Threads.

Table 6. Supported Components

Variable Name	Default Value
THRD	Threading, Thread Pools
STCK	The modular stack and how operations propagate up and down
BSCN	operations related to the blob store connection
STAT	Deals with States
MPAU	Multi-part upload
CNCT	Connectors
SLCE	Slices

Table 6. Supported Components (continued)

Variable Name	Default Value
KMGR	Key Management
ENCR	Encryption
GCON	Scale back end connector
ETAG	Etag Based Integrity
MFST	Manifest related Operations
PAYL	Payload related Operations
ENVR	Environment related operations
CDIR	Cloud Directory Component Operations
RECN	Reconcile
SCAN	Scan operations
POLI	GPFS Policy operations
AUTH	Authentication
SQLL	SQL operations
JRNL	Journal operations
NOTF	GPFS Event Notifications
MTRX	Metrics Code
CDAM	Core Data Model
GDAM	GPFS Data Model
MTTV	TCT TTV Validator
CONF	TCT configuration Layer related operations.
SERV	Serviceability
MMON	Monitoring

To tune Cloud services parameters, issue a command according to this syntax:

```
mmcloudgateway config set --cloud-nodeclass CloudNodeClass
                           Attribute=value[,Attribute=value...]
```

where,

- <--cloud-nodeclass> is the node class configured for the Cloud services nodes.
- <Attribute> is the value of the attribute that is provided.

For more information, refer to the following examples:

- To set the value of the `tconnector.server.timeout` attribute to 10 seconds, issue this command:

```
mmcloudgateway config set --cloud-nodeclass tct1 connector.server.timeout=10000
```

- To reset the value of the `tconnector.server.timeout` attribute to the default value, issue this command:

```
mmcloudgateway config set --cloud-nodeclass tct1 connector.server.timeout=DEFAULT
```

- To set the tracing levels, issue this command:

```
mmcloudgateway config set --cloud-nodeclass tct tracing.level=GCON=5:AUTH=4
```

- To reset the value of the tracing components to the default value, issue this command:

```
mmcloudgateway config set --cloud-nodeclass tct tracing.level=default
```

- To reset the values of multiple attributes, issue this command:

```
mmcloudgateway config set --cloud-nodeclass tct tracing.level=CDIR=1:  
JRNL=1:SQLL=1:RECN=1
```

Integrating Cloud services metrics with the performance monitoring tool

This topic describes the procedure for integrating Cloud services server nodes with the performance monitoring tool.

Performance monitoring collector (pmcollector) and sensor (pmsensors) services are installed under /opt/IBM/zimon.

Note: You must install the sensors on all the cloud service nodes, but you need to install the collectors on any one of the GPFS nodes. For more information about the installation instructions, see *Manually installing the performance monitoring tool* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Two types of configurations are possible for the performance monitoring tool integration:

- GPFS-based configuration (configuration by using GPFS commands). This type is recommended.
- File-based configuration (manual configuration of the sensor files as described here).

For more information, see *Configuring the performance monitoring tool* in *IBM Storage Scale: Problem Determination Guide*.

GPFS-based configuration

This topic describes the procedure for integrating cloud service metrics with the performance monitoring tool by using GPFS-based configuration.

1. On the Cloud services nodes, copy the following files from /opt/ibm/MCStore/config folder to /opt/IBM/zimon folder:

- TCTDebugDbStats
- TCTDebugLweDestroyStats
- TCTFsetGpfsConnectorStats
- TCTFsetIcstoreStats
- TCTFsGpfsConnectorStats
- TCTFsIcstoreStats

2. Register the sensor in the GPFS configuration by storing the following snippet in the MCStore-sensor-definition.cfg file:

```
sensors=  
{  
    # Transparent cloud  
    tiering statistics  
    name = "TCTDebugDbStats"  
    period = 10  
    type = "Generic"  
},  
  
{  
    #Transparent cloud
```

```

tiering statistics
  name = "TCTDebugLweDestroyStats"
  period = 10
  type = "Generic"
},
{
  #Transparent cloud
tiering statistics
  name = "TCTFsetGpfssConnectorStats"
  period = 10
  type = "Generic"
},
{
  #Transparent cloud
tiering statistics
  name = "TCTFsetIcstoreStats"
  period = 10
  type = "Generic"
},
{
  #Transparent cloud
tiering statistics
  name = "TCTFsGpfssConnectorStats"
  period = 10
  type = "Generic"
},
{
  #Transparent cloud
tiering statistics
  name = "TCTFsIcstoreStats"
  period = 10
  type = "Generic"
}

```

3. Run this command:

```
prompt# mmperfmon config add --sensors MCStore-sensor-definition.cfg
```

Note: The sensor definition file can list multiple sensors separated by commas (,).

For more information on GPFS-based configuration, see the topic *mmperfmon command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

File-based configuration

This topic describes how to configure Cloud services with the performance monitoring tool by using file-based (manual) configurations.

Note: You must delete the sensors that are used in the earlier releases. If the scope of your Cloud services configuration is file system, then you do not need to configure the sensor files that start with *TCTFset**. Similarly, if the scope of your Cloud services configuration is fileset, then you do not need to configure the sensor files that start with *TCTFs**.

To integrate the performance monitoring tool with Cloud services server nodes, do the following steps:

1. Copy /opt/IBM/zimon/defaults/ZIMonSensors.cfg to /opt/IBM/zimon. This configuration file determines which sensors are active and their properties.

Note: If the sensors are already configured at /opt/IBM/zimon/defaults/ZIMonSensors.cfg, use the same sensors.

2. Edit the /opt/IBM/zimon/ZIMonSensors.cfg file and set an IP address for the “host” attribute in the “collectors” section.

Note: If the collectors are already configured at /opt/IBM/zimon/ZIMonSensors.cfg, use the same collectors.

3. Edit the /opt/IBM/zimon/ZIMonSensors.cfg file to append the following sensors at the end of the sensor configuration section:

```

sensors=
{
    # Transparent cloud
tiering statistics
    name = "TCTDebugDbStats"
    period = 10
    type = "Generic"
},
{
    #Transparent cloud
tiering statistics
    name = "TCTDebugLweDestroyStats"
    period = 10
    type = "Generic"
},
{
    #Transparent cloud
tiering statistics
    name = "TCTFsetGpfsConnectorStats"
    period = 10
    type = "Generic"
},
{
    #Transparent cloud
tiering statistics
    name = "TCTFsetIcstoreStats"
    period = 10
    type = "Generic"
},
{
    #Transparent cloud
tiering statistics
    name = "TCTFsGpfsConnectorStats"
    period = 10
    type = "Generic"
},
{
    #Transparent cloud
tiering statistics
    name = "TCTFsIcstoreStats"
    period = 10
    type = "Generic"
}

```

Note: Each sensor should be separated by a comma. The period is the frequency in seconds at which the performance monitoring tool polls the cloud service for statistics. The period is set to 10 seconds but it is a configurable value. The sensor is turned off when the period is set to 0.

4. Copy the following files from /opt/ibm/MCStore/config folder to /opt/IBM/zimon folder:

- TCTFsGpfsConnectorStats.cfg
- TCTFsIcstoreStats.cfg
- TCTFsetGpfsConnectorStats.cfg
- TCTFsetIcstoreStats.cfg
- TCTDebugLweDestroyStats.cfg
- TCTDebugDbStats.cfg

5. Restart the sensors by using this command: **service pmsensors restart**.

6. Restart the collectors by using this command: **service pmcollector restart**

Note: If the collector is already installed and is running, then only pmsensors service needs to be restarted. If you are installing both pmcollectors and pmsensors, then both services need to be restarted.

Setting up Transparent cloud tiering service on a remotely mounted client

When you use Transparent cloud tiering on a remote cluster, you must complete your administrative work and policy migration scheduling on the Transparent cloud tiering server nodes. User access to tiered data from remotely mounted clusters is available by setting up transparent recalls (which is outlined here).

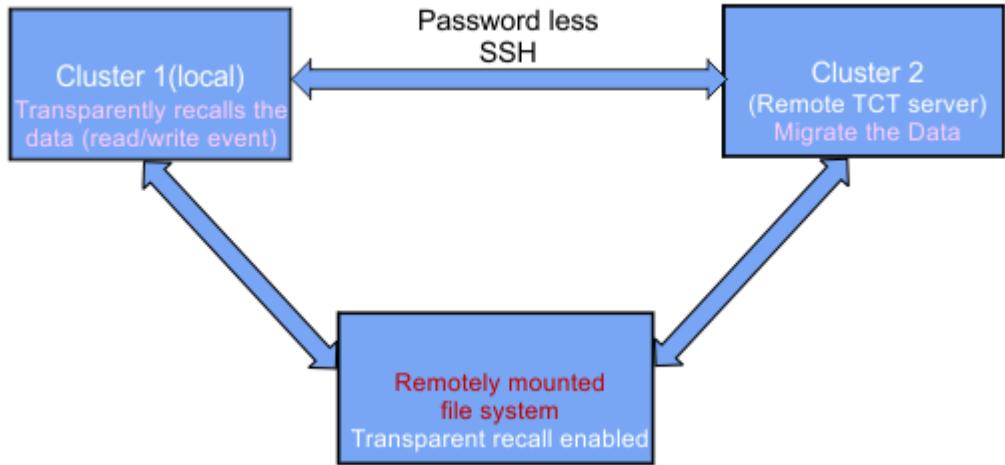


Figure 5. Transparent cloud tiering on a remote cluster

Before you begin, ensure the following:

- A local cluster and a remote cluster are created. For more information, see the *mmcrfs command* in the *IBM Storage Scale: Command and Programming Reference Guide*.
 - A passwordless SSH is set up between the local and the remote clusters. For instructions on passwordless SSH setup and other prerequisites, see the *Prompt-less SSH setup* section of the *Problems due to missing prerequisites* topic in the *IBM Storage Scale: Problem Determination Guide*.
 - Transparent cloud tiering server RPMs are installed on the local cluster.
1. Create authentication between the local cluster and the remote cluster. For more information, see the *Mounting a remote GPFS file system* topic in the *IBM Storage Scale: Administration Guide*.
 2. Verify that the file system is mounted on the remote cluster (from the local cluster) by issuing the following command on the remote cluster:

```
mmremotecluster show
```

The system displays output similar to this:

```
Cluster name: vm1.pk.slabs.ibm.com
Cluster id: 5646242228948715502
Contact nodes: vm1.pk.slabs.ibm.com
SHA digest: 37a856428f565d017gg4abb935a81493d66cd0498917e8ef750c1b5e4bc60d56
SHA digest: mygpfsnew (gpfs0)
```

3. On the local cluster, set the following GPFS variable, which will point to the gateway node of the remote cluster that needs to be connected for transparent recall:

- a. For a single remote cluster mounted, set the variable, as follows:

```
mmcr vput tct<remote_cluster_name> <IP_address>
```

For example, to set the GPFS variable on local cluster for remote cluster tctvm1.pk.slabs.ibm.com, issue the following command:

```
mmccr vput tctvm1.pk.slabs.ibm.com 192.0.2.0
```

b. For multiple remote cluster mounted, set the variable, as follows:

```
mmccr vput tct<remote_cluster1_name> <IP_address>
mmccr vput tct<remote_cluster2_name> <IP_address>
```

For example, to set the GPFS variable on local cluster for two remote clusters, tctvm1.pk.slabs.ibm.com and tctvm2.pk.slabs.ibm.com, issue the following command:

```
mmccr vput tctvm1.pk.slabs.ibm.com 198.51.100.1
mmccr vput tctvm2.pk.slabs.ibm.com 198.51.100.2
```

4. Copy the `mcstore` script from the local cluster to the remote cluster by issuing the following command:

```
cp /opt/ibm/MCStore/scripts /opt/ibm/MCStore/bin/
```

5. Migrate data from the local cluster to the cloud by using the `mmcloudgateway files migrate` command.

6. Transparently recall data from the cloud (by issuing the `cat` or any similar command on the remote cluster CLI).

Deploying WORM solutions

This topic describes how you can set up a WORM (write once and read many) solution by using IBM Storage Scale, Transparent cloud tiering, and IBM Cloud Object Storage.

IBM Storage Scale provides the immutability feature where you can associate a retention time with files, and any change or deletion of file data is prevented during the retention time. You can configure an IBM Storage Scale fileset with an integrated Archive Manager (IAM) mode by using the `mmchfileset` command. Files stored in such an immutable fileset can be set to immutable or append-only by using standard POSIX or IBM Storage Scale commands. For more information on immutability features available in IBM Storage Scale, see [“Immutability and appendOnly features” on page 594](#).

After immutability feature is configured in IBM Storage Scale, you can ensure that files that are stored on the Object Storage are immutable by leveraging the locked vault feature available in IBM Cloud Object Storage.

Locked vaults enable storage vaults to be created and registered under the exclusive control of an external gateway application. IBM Cloud Object Storage stores objects received from the gateway application. The gateway authenticates to the IBM Cloud Object Storage Manager exclusively by using an RSA private key and certificate that was configured to create a locked vault and registered only with the gateway. After that, the normal S3 APIs can be used against the Accesser nodes by using the configured private key and certificate. Accesser API key and secret key for S3 API cannot be used for authentication or authorization.

If a key is compromised, the gateway rotates keys by calling the Rotate Client Key Manager REST API. This API replaces the existing key and revokes the old certificates. A locked vault with data cannot be deleted by the IBM Cloud Object Storage Administrator, and its ACLs cannot be changed. Additionally, it cannot be renamed or have proxy setting enabled. For more information about locked vaults, see *IBM Cloud Object Storage System Locked Vault Guide*.

Note: To configure WORM feature at the fileset level, it is recommended to match the immutable filesets with immutable container pair sets on the cloud.

Creating immutable filesets and files

To configure and deploy a WORM solution, it is mandatory to create an immutable fileset in IBM Storage Scale.

1. Create an independent fileset by using the **mmcrfileset** command.
2. Link the fileset to a directory within the file system which must not exist at this point:

```
mmlinkfileset <file system name> <fileset name> -J directory
```

Note: This directory is the immutable fileset path.

3. Set an IAM mode for the files by using the following command:

```
mmchfileset <file system name> <fileset name> --iam-mode compliant
```

4. Create a test file `date > testfile` with read-write permissions and fill the file with some content:

```
echo "Hello World" > file
```

5. Check the extended attributes of the file which indicate that the file is not immutable by using the **mmlsattr** command:

```
[root@localhost WORMfs]# mmlsattr -L testfile
file name: testfile
metadata replication: 1 max 2
data replication: 1 max 2
immutable: no
appendOnly: no
indefiniteRetention: no
expiration Time: Wed Mar 15 17:16:13 2016
flags:
storage pool name: system
fileset name: WORMfs
snapshot name:
creation time: Wed Mar 15 17:16:13 2016
Misc attributes: ARCHIVE
Encrypted: no
```

6. Set the file to read-only:

```
chmod -w testfile
```

7. Set the future expiration time using **mmchattr**. Select a time in the immediate future for quick expiry and deletion.

```
mmchattr --expiration-time 2016-03-15@18:16:13 testfile
```

8. Verify that immutability and expiration time are set by using **mmchattr**:

```
mmlsattr -L testfile
file name: testfile
metadata replication: 1 max 2
data replication: 1 max 2
immutable: yes
appendOnly: no
indefiniteRetention: no
expiration Time: Wed Mar 15 18:16:13 2016
flags:
storage pool name: system
fileset name: WORMfs
snapshot name:
creation time: Wed Mar 15 17:16:13 2016
Misc attributes: ARCHIVE READONLY
Encrypted: no
```

9. Verify that the files cannot be modified or deleted. Run the following commands:

```
# chmod +w testfile
```

The system displays an output similar to this:

```
chmod: changing permissions of 'testfile': Operation not permitted  
# date > testfile
```

The system displays an output similar to this:

```
testfile: Read-only file system  
# rm -f testfile
```

The system displays an output similar to this:

```
rm: cannot remove 'testfile': Read-only file system
```

For more information, see the following topics:

- “[Immutability and appendOnly features](#)” on page 594
- <https://www.ibm.com/support/pages/node/6355547>

Setting up Transparent cloud tiering for WORM solutions

After an immutable fileset is configured, you must set up Transparent cloud tiering to be able to configure and deploy WORM solutions. This topic describes the procedure for doing that.

Before you begin, ensure the following:

- IBM Cloud Object Storage version is at 3.9.x or later.
- IBM Cloud Object Storage is run on the "vault" mode, not on the "container" mode.
- To run the scripts, you have to create a private account on the IBM Cloud Object Storage.

Note: You can perform these procedures either manually or by using the scripts available in the package.

The following scripts are available at /opt/ibm/MCStore/scripts:

- mcstore_createlockedvault.sh
- mcstore_lockedvaultpreconfig.sh
- mcstore_lockedvaultrotateclientkey.sh

1. Set up a private key and create locked vaults.
2. Configure Transparent cloud tiering with certificate-based authentication and locked vaults.
3. Rotate client key or revoke old certificate.
4. Update Transparent cloud tiering with new private key and certificate.

Setting up a private key and creating locked vaults

The first step involves creating a certificate signing request (CSR), registering certificate with IBM Cloud Object Storage via Client Registration REST API and obtaining a private key (RSA based).

Once the private key is obtained, it can be used to create the locked vaults on the IBM Cloud Object Storage system. Additionally, for HTTPS (TLS), the CA certificate of the IBM Cloud Object Storage system is also required.

The two locked vaults required for Transparent cloud tiering (data and metadata vaults) need to be created on the IBM Cloud Object Storage by using the *create vault from template* REST API. Once these vaults are created, they can be specified on the **mmcloudgateway filesystem create** command via the –container-prefix option.

Setting up a private key and a private certificate

This topic describes the procedure for setting up a private key and a private certificate for deploying WORM solutions by using IBM Cloud Object Storage.

The first step involves creating a certificate signing request (CSR) and registering the client certificate with IBM Cloud Object Storage Manager via Client Registration REST API and obtaining a private key (RSA based) signed with IBM Cloud Object Storage Manager Certificate Authority. Once the signed private certificate is obtained, we can use the RSA private key and private certificate for creating the locked vaults on the IBM Cloud Object Storage system. Additionally, for HTTPS (TLS) communication, the root CA certificate of the IBM Cloud Object Storage system is also required.

Note:

- A private account must be created before an automation script or procedure is run.
- A private account must be created each time an incorrect IBM Cloud Object Storage CA certificate is specified while generating the Keystore.

1. Create a directory that will hold private key and certificates by issuing this command:

```
$mkdir mydomain2.com.ssl/
```

2. To generate a keystore that will store the private key, CSR, and certificates, issue the following command in the /opt/ibm/MCStore/jre/bin directory:

```
keytool -genkey -alias mydomain2 -keyalg RSA -keysize 2048 -keystore mydomain2.jks
```

Note: You should make a note of the alias name as it has to be used in the later steps.

3. Generate CSR by issuing the following command:

```
keytool -certreq -alias mydomain2 -keyalg RSA -file mydomain2.csr -keystore mydomain2.jks
```

4. Create a private account on IBM Cloud Object Storage Manager.

5. Using the private account created, send the CSR to IBM Cloud Object Storage Manager to be signed by issuing the following command:

```
curl -u <privateuser>:<password>
-k 'https://<COS Manager IP>/manager/api/json/1.0/clientRegistration.adm'
-d 'expirationDate=1508869800000' --data-urlencode 'csr-----BEGIN NEW CERTIFICATE REQUEST-----
MIICzjCCAbYCAQAwWTELMAkGA1UEBhMCSU4xCzAJBgNVBAgTAKtBMRIwEAYDVQQHEw1CYW5nYWx
cmUxDDAKBgNVBAoTA1NEUzENMASGA1UECxMESVNETDEMMaG1UEAxMDSUJNMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIBCgKCAQEcApfVgjn9p9vBwGA6Y/g54DBr1wWtWeSAwm680M4201PUuRwV92
9UDBK9xEkY2Zb+o08Hvspd5VMU97bV7cnN8Fii8WuujHCdgaVuezTT0ZCHjVH12L6CYq17hmWIazk
T0aR0oY1hzZCgQrDyVNiw6XuvkWo3eUIRyi1r6naUFiqUtMeerEhEYa6cmm5qpeb2GKYJdeN53W
SF0yrUCi9gRgPjiaQ6lVS1+wWekbI6lwIA+jVyojx931RL/KdxFfmh/sriUx//a6+I00Bl6EmEV
BsHeG2HccS1diJJ4+eUetXvfkyMj06kRvYraSVKX022a4Jqki8iYDNf4XvRzOz5YbLQIDAQABoDAw
LgYJKoZIhvCNQkOMSEwHzAdBgNVHQ4EFgQUrgpT7F8Z+bA9qDxqU8Pd70zFj4wdQYJKoZIhvCN
AQELBQADggEBADW4xuxBaaH9/ZBL0110tXveSHF8Q4oZo2MhsSwf34Shu/Zxc17H8NqCCMyxqVdXI
6kb6dg5se5WLcQ/JJA7TBcgCyJJqVjADt+RC+TGNC0N1sC7XpeRYLJtxq1KilsWnKJf5oRvA1Vg5P
nkTjCE9XvUzhJ/tTQjNBJS8nN7Tbu/q5mTIGG9imARPro2xQpvwiFMHrq/f1uNeZ3SeuLxwQtK
4zge7XwyY63l1RsN0z2a4CPNzU0q68TGL1aE930DpJYuSeTB0m2om4iTNSngsQKRmYqGDSXM3no/
90uTeTAGhjhJ82bgEOfP9FVm+6FnYydr1Endg1aEizC+sAirk4e8E=
-----END NEW CERTIFICATE REQUEST-----' -v
```

Note: The expiration time should be specified in milliseconds.

6. Curl command provides a certificate in the response, as follows:

```
"-----BEGIN CERTIFICATE-----
\nMIIEczCCAlugAwIBAgIQeijQBskfm0v3kYQcBOBmxTANBgkqhkiG9w0BAQ0FADCB\nkTELMAkGA1UE
BhMCVVMxETAPBgNVBAgMCE1sbGlub21zMRAwBgYDVQQHAdDaG1j\nnYWdvMRMwEQYDVQQKDApDbGV2ZX
JzYWZ1MRkwFwYDVQDDBBk051dCBNYW5hZ2Vy\nnIENBMS0wKwYDVQQFEyQwMmQxMjk5ZS05Nzc3LTR1
NmItODg3Yy0wYmNzJkODU1\nnMzcvhcnNMTyxMDI0MTMxNTE2WhcNMTxMDI0MTgzMDAwWjBZMQswCQ
YDVQGEwJJ\nnTjELMakGA1UECBMCS0ExEjAQBgNVBAcTCUjhbmdbG9yZTEMMaoGA1UEChMDU0RT\nnMQ
0wCwYDVQQLEwRJU0RMQMwwCgYDVQQDeNjQk0wggEiMA0GCSqGSIb3DQEBAQUA\nnA4IBDwAwggEKAoIB
AQCI9WC0en28HAYDpj+DngMGvxBa1Z5IDCbzrQzjY7U9S5HB\nnX3b1QMEr1cSRjZ1v6jTwe+y131UxT3
ttXtyc3wWLxa66McJ2ABW57NNPRkIeNUeX\nnYvoJiqXuGZYhr0RM5pE6hiWHNkKBCsPJU0jDpe6+Rajd
```

```

5QhHKLWvqdp9QWKpS0wR\n6sSERhryabmq15vYYpgl143ndZIXTktQKL2BGA8mICrqVVKX7BZ6RsjqX
AgC01X\nKiPH3eVGX8p3F8WaH+yuJTH/9rr4j04GWLoSYRUGwd4bYdxxLV2Inj55R61e9+Rg\nyM7qRG
91tpJUpfTbZrgmqSLyJgM1/he9HM7P1hstAgMBAAEwDQYJKoZIH* vcNAQEN\nnBQADggIBAJmCnhIN/
nhp2V1gqA7td3EBD8xrejF0btTmSUgx8f1FmCKCJh6/0yn9\n11PUp3SzSu734GdTZiUTXax7PYZ1B
3ST1Y0sZE7yU6za101IoUZEzXoohIEPVU\nw4X3j9HF3hWDNsuzqZfQDRmdaz6NG2EPDxiWgTYXPLdY
aZyTQFFe6A4tbT9gSHu\nn9UD1woFwjrsAfg03zwR7wsRSwcAlsVs1BK96TyufZf+E2eFg+QBGA5YWrZ
i3g40\n1Xqxj5W5TwujLxSJ+8zx6P9f0T96VGICH8Yy9AIWzUa3fXlh6tc1Pw+LbuIjEWr\nnK2TS+DL
TmBAo8p05GsR8rShKFcPY0ho2mbskAKgt4n+s63Jhu5qALS4Lw7eEQ7W7\nnqGffZ2JttNHwePAqvxx33
xf+Y2Swn0fb0AlwT9BQ6ySn/qZ3e3Xl0rVqqukgCq0\nnBn0H15WN4Hk0NkyaquJruTLHUIWX5T01q/y
LnzRt8TCBA4qnX7HMLEmQkXiF5Poj\nBcyCTctYu1H1ijHjsW09kztUfljI50kVyS1q1FqcZQiziHHRI
AEWbnrYn6Fgq13g\nIws7Lw9Utogj54tPCwJ8gEkoW4eT04tnZmPTTdWlmVhTdEjVRxE8fotztHJuVis
P\ncFxBPWJZ8IP9t2C/4Zi1PuqXI/8YZx8LPicQuCrxelURigQrbp7
\n-----END CERTIFICATE-----\n"

```

7. Remove the '/n' character from the certificate (from BEGIN to END CERTIFICATE) and store the certificate in a file.
8. Get the CA certificate of IBM Cloud Object Storage Manager and import into the keystore created in step 2. To import the CA certificate, issue the following command:

```

keytool -importcert -trustcacerts -noprompt -alias cleversafeeca -file
<cleversafe-cafile-loc> -keystore mydomain2.jks -storepass <keystore-password>

```

9. Import the certificate into the keystore by issuing the following command:

```

keytool -importcert -trustcacerts -alias mydomain2 -file <client-cert-location> -keystore
mydomain2.jks -storepass <keystore-password>

```

Note: You can set up a private key and a private certificate by using this script **mcstore_lockedvaultpreconfig.sh** available at /opt/ibm/MCStore/scripts, as follows:

Setting up a private key and private certificate by using the automation script

- a. Run **mcstore_lockedvaultpreconfig.sh** <keystorealiasname> <keycertLocationDirectory> <COSManagerIP> <username> <expirationDays> <COSCACertFile>, where the first 4 arguments are mandatory and the last two (expirationDays and COSCACertFile) are optional.

If the expiration date (expirationDays) is not specified, then the command will take the default expiration time, which is 365 days.

If the IBM Cloud Object Storage CA certificate (COSCACertFile) is not specified, then the CA file will be downloaded from the IBM Cloud Object Storage Manager.

- b. For more information on the description of the parameters, see the **mmcloudgateway** man page.

For example,

```
./mcstore_lockedvaultpreconfig.sh test /root/svt 9.10.0.10 newuser2
```

The system displays output similar to this:

```

Enter KeyStore Password:
Enter Private Account Password:
Validating the inputs and the configuration...
COS Manager is reachable. Proceeding with Configuration...

Transparent Cloud Tiering Server RPM already installed. Proceeding with Configuration...

Python libraries are already installed. Proceeding with Configuration...

CURL already installed. Proceeding with Configuration...

Downloading COS CA Certificate....
Validation completed for inputs and the proceeding with configuration....
Generating a new Keystore and Private Key...
What is your first and last name?
[Unknown]: dmeo1
What is the name of your organizational unit?
[Unknown]: dmeo1
What is the name of your organization?
[Unknown]: demo2
What is the name of your City or Locality?

```

```

[Unknown]: demo1
What is the name of your State or Province?
[Unknown]: demo
What is the two-letter country code for this unit?
[Unknown]: KA
Is CN=dmeo1, OU=dmeo1, O=demo2, L=demo1, ST=demo, C=KA correct? (type "yes" or "no")
[no]: yes

Importing COS CA Certificate to Key Store.....
Certificate was added to keystore
Generating a CSR.....
Sending CSR to CleverSafe to be signed.....
  % Total    % Received % Xferd  Average Speed   Time     Time     Time  Current
               Dload  Upload   Total   Spent    Left  Speed
100  2990  100  1781  100  1209  5310  3605 --::-- --::-- --::--  5316
Retrieving Certificate from Response.....
Importing Client Certificate to Keystore.....
Certificate reply was installed in keystore
Pre-configuration for Locked Vault completed successfully.
IMPORTANT: /root/svt/test.ssl contains private key, keystore and private certificate.
You must keep a back up of /root/svt/test.ssl.

```

Creating locked vaults

To deploy WORM solutions by using IBM Storage Scale, create two locked vaults.

IBM Cloud Object Storage Manager enables administrators to create vaults, which are under the exclusive control of a given external application (Transparent cloud tiering). This process allows the application to have full control over the vault, but does not allow a user or administrator to bypass the application and directly access the vault. Users are allowed to create WORM-style vaults that enforce read or write restrictions on the objects in the vault, which an administrator cannot bypass.

The two locked vaults required for Transparent cloud tiering (data and metadata vaults) need to be created on the IBM Cloud Object Storage by using Create vault from the template REST API. When these vaults are created, they can be specified on the **mmccloudgateway filesystem create** command through the –container-prefix option.

Note: You can create a locked vault by using the `mcstore_createlockedvault.sh` script available at `/opt/ibm/MCStore/scripts`.

- Convert the JKS keystore to the PKCS12 format by issuing this command:

```
keytool -importkeystore -srckeystore mydomain2.jks -destkeystore new-store.p12
-deststoretype PKCS12
```

- Extract the private key and convert it to an RSA key by issuing the following commands:

- `openssl pkcs12 -in "<keystore_directory>"/newkeystore.p12 -nocerts
-out "<keystore_directory>"/privateKey.pem -passin pass:<keystore_password>
-passout pass:<keystore_password>`
- `openssl rsa -in "<keystore_directory>"/privateKey.pem -out "<keystore_directory>"/
rsaprivateKey.pem
-passin pass:<keystore_password>`

- By using the private key and certificate, create a locked vault (one for data and one for metadata) by issuing the following commands:

- For data vault:

```
curl --key ./ privateKeynew.pem --cert <certificate-file> -k -v
'https://9.114.98.187/manager/api/json/1.0/createVaultFromTemplate.adm'
-d 'id=1&name=demolockedvault&description=newlockedvaultdescription'
```

- For metadata vault:

```
curl --key ./ privateKeynew.pem --cert <certificate-file> -k -v
```

```
'https://9.114.98.187/manager/api/json/1.0/createVaultFromTemplate.adm'  
-d 'id=1&name=demolockedvault.meta&description=newlockedvaultmetadescription'
```

Note: To find the provisioning template IDs, on the IBM Cloud Object Storage Manager GUI, click **Template Management**. Then, hover the mouse over the template that is listed under **Vault Template**, and find the number that is displayed on the footer.

4. Print the locked vaults by issuing this command:

```
curl --key privateKeynew.pem --cert <certificate-file> -k '<COS Accesser IP Address>'
```

Note: The names of the locked vaults must be noted down, and they must be specified to the **mmcloudgateway filesystem create** command by using the **--container-prefix** option.

Creating a locked vault by using automation scripts

- a. Go to /opt/ibm/MCStore/scripts and run

```
mcstore_createlockedvault.sh <keystorealiasname> <keyStorePath>  
<lockeddatavaultname> <lockeddatavaultDescription> <lockedmetavaultname>  
<lockedmetavaultDescription> <COSManagerIP> <dataVaultTemplateID>  
<metaVaultTemplateID>, where all parameters are mandatory.
```

- b. For description of the parameters, see the **mmcloudgateway** command.

```
For example, mcstore_createlockedvault.sh test /root/svt/test.ssl/test.jks  
demodatacontainer test demodatacontainer metacontainer 9.10.0.10 1 1.
```

The system displays output similar to the example shown:

```
Enter KeyStore Password:  
Validating the inputs and the configuration....  
COS Manager is reachable. Proceeding with Configuration...  
  
Transparent Cloud Tiering Server RPM already installed. Proceeding with Configuration...  
openssl libraries are already installed. Proceeding with Configuration...  
  
curl already installed. Proceeding with Configuration...  
Certificate stored in file </root/svt/test.ssl/test_new.crt>  
Creating locked vault...  
MAC verified OK  
writing RSA key  
Locked data vault creation completed successfully.  
Creating locked meta vault demofeb15.meta  
  
Creating of Data and Meta Locked Meta Vault completed successfully.  
Use mmcloudgateway filesystem create command to configure transparent cloud tiering  
with locked vault.
```

Configuring Transparent cloud tiering with certificate-based authentication and locked vaults

This topic provides how to configure Transparent cloud tiering with certificate-based authentication and locked vaults.

Be sure to keep a backup copy of the source keystore that you used to import the private key and certificates. The **mmcloudgateway account delete** command removes the private key and certificates from the trust store.

1. Get the client certificate for IBM Cloud Object Storage Accesser.
2. Create a cloud storage account by using the **mmcloudgateway account create** command. For more information, see [“Managing a cloud storage account” on page 90](#).
3. Create a cloud storage access point (CSAP) by using the **mmcloudgateway containerPairSet create** command. For more information, see [“Binding your file system or fileset to the Cloud service by creating a container pair set” on page 96](#).
4. Create a cloud service by using the **mmcloudgateway cloudservice create** command. For more information, see [“Creating Cloud services” on page 94](#).

5. Configure Cloud services with SKLM by using the **mmcloudgateway keyManager create** command. For more information, see “[Configuring Cloud services with SKLM \(optional\)](#)” on page 95.
6. Create a container pair set by using the **mmcloudgateway containerpairset create** command. For more information, see “[Binding your file system or fileset to the Cloud service by creating a container pair set](#)” on page 96.
7. Perform migrate and recall operations by using commands or policies.

Rotating Client Key or revoking old certificate

Once the client key is rotated, you must use the new certificate and private key to be able to create locked vaults. You can perform this procedure by using the following steps or by using this script: /opt/ibm/MCStore/scripts/mcstore_lockedvaultrotateclientkey.sh.

Note: Before you perform this procedure, ensure that no active migration is currently taking place. After you perform this procedure, the old keys will not work.

1. Generate a new CSR using a new alias:

```
keytool -certreq -alias mydomainnew -keyalg RSA -file mydomainnew.csr -keystore mydomain2.jks
```

2. Get the CSR signed by sending it to the IBM Cloud Object Storage Manager:

```
curl --cacert {path to ca certificate} --key {path to RSA private key}
--cert {path to old certificate}
'https://<Cloud Object Storage Manager IP>/manager/api/json/1.0/rotateClientKey.adm'
-d 'expirationDate=1508869800000' --data-urlencode 'csr=
-----BEGIN NEW CERTIFICATE REQUEST-----
MIICzjCCABYCAQAwTELMAkGA1UEBhMCSU4xCzAJBgNVBAgTAKtBMRIwEAYDVQQHEw1CYW5nYWxv
cmUxDAAKBgNVBAoTA1NEUzENMasGA1UECxMESVNETDEMMa0GA1UEAxMDSUJNMIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEApfVgjnp9vBwGA6Y/g54DBr1wWtWeSawm680M4201PUuRwV92
9UDBK9XEkY22b+o08Hvspd5VMU97bV7cn8Fl8WuujHCdgAVuezTT0ZChjVH12L6CYq17hmWIazk
TOaR0oYlhZCgQrDyVNIw6XuvkWo3eUIRiy1r6naUFiqUtMEerEhEY46cm5qpeb2GKYJdeN53W
SF0yrUCi9gRgPjiAq61VS1+wWekbI6lwIAtJyvjojx931R1/KdxFmh/sriUx//a6+I0OB1i6EmEV
BsHeG2HccS1diJ4+eUetXvfkYMj06kRvYraSVKX022a4Jqki8iYDNf4XvRzOz5YbLQIDAQABoDAw
LgYJKoZIhvvcNAQk0MSEwHzAdBgNVHQ4EfQURgpT7F8Z+bA9qDxqU8PDg70zFj4wDQYJKoZIhvvcN
AQELBQADggEBADW4xuxBaaH9/ZBL01l0tXvSeHF8Q4oZo2MhSwf34Shu/ZxC17H8NqCCMyxqVdXI
6kbgd1se5WLcQ/jJA7TBcgCyJqVjADt+R+TGNc0NlsC7XpeRYLtxqlKilsWnKJf5oRvA1Vg5P
nkTjCE9XvUzhJ/tTQjNBJS8nN7Tbu/q5mTiGG9imARPro2XQpvwiFMHrq/f1uNeZ3SeuLxwQtK
4zge7XwyY631rKsN0z2a4CPNzU0q68TGL1aE93QDpJYusSeTB0m2om4iTSNgSQRmYqGDSXM3no/
90UeTAGhjhJ82bGEOfP9FVm+6FnYydr1Endg1aEizC+sAirk4e8E=
-----END NEW CERTIFICATE REQUEST-----' -v
```

3. Curl command provides a new certificate, as follows:

```
"-----BEGIN CERTIFICATE-----
\nMIIEczCCAlugAwIBAgIQejjQBskfm0v3kYQcB0BmxTANBgkqhkiG9w0BAQ0FADC
\nkTELMAkGA1UEBhMCVVMxETAPBgNVBAgMCE1sbGlub21zMRawDgYDVQQHDAadDaG1j
\nnYldvMRMwEQYDVQKDApDbGV2ZXJzYwZ1MRkwFwYDVQQDBBk051dCBNYW5hZ2Vy
\nnIEBMS0wKwYDVQFEEyQwMmQxMjk5ZS05Nzc3LTr1NmItODg3Yy0wYmMzNzjk0DU1
\nnMzcwHhcNMTYxMDI0MTMxNTE2WhcNMTCxMDI0MTgzMDAwWjEZMQswCQYDVQGEwJJ
\nnTjELMAkGA1UECBMCSoExEjAQBgNVBAcTCUjhbmdbG9yZTEMMa0GA1UEChMDU0RT
\nnMQ0wCwYDVQLEwRJU0RMQwwCgYDVQDDeWJQk0wggEiMA0GCSqGSIB3DQEBAQUA
\nnA4IBDwAwggEKAoIBACQ1wCoen28HAYDpj+DngMGvXBa1Z5IDCbrzQzjy7U9S5HB
\nnX3b1QMEr1cSRjZl6jtWe+y131UxT3ttXtyc3wWLxa6McJ2ABW57NNPRkIeNUeX
\nnYv0JiqXuGZYhi0RM5pE6hiWHNkKBCsPJU0jDpe6+Rajd50HHLWvqdp90WkpS0wR
\nn6sSERhpyabmq15vYYpgl143ndZIXTkTqKL2BGA8mICrqVVKX7BZ6RsjqXAgC01X
\nnK1PH3eVGX8p3f8WaH+yuJTH/9rr4j04GWLsSYRUGwd4bYdxLV2Inj55R61e9+Rg
\nnyM7qRG9itpJUpfTbZrgmcSLyJgM1/he9Hm7PlhstAgMBAAEwDQYJKoZIh*vCNAQEN
\nnBQAoggIBAJmCnhIN/nhp2VIggA7td3EBD8xrejF0bT5mSUgx8f1FmCKCJh6/Oyn9
\nn11PUp3SzSu734GdTdzIUtTXax7PYZ1B3ST1Y0sZE7yU6zal01IoUZEzXoohIEPVU
\nnW4X3j9HF3hWDwNsqqZfQDRmndaz6NG2EPDxiWgTYXPLdYaZyTQFFe6A4tbT9gSHu
\nn9UD1woFwjrsAf03zwR7wSRSwcALsVs1BK96TYufZf+E2eFg+QBGAC5YwZi3g4Q
\nn1Xqjx5W5TwuJLxSJ+8zxj6P9f0t96VGICH8Yy9AIWzUa3fxLh6tc1Pw+LbuIjeWr
\nnK2Ts+DLTmBa08p05GsR8rShKFcPY0h02mbskAGt4n+s63Jhu5qALS4Lw7eEQ7W7
\nnqGffZ2JttNHwePAaqvx33xf+Y2SwN0fb0A1wT9BQ6ySn/gZr3e3X10rVqqukgCq0
\nnBnQhI5WN4HkOnkyaquJruTLHU1WX5T01q/yLnrRt8TCBA4qnX7HMLEmQkXiF5Poj
\nnBcyCTctYu1HijHjsW09kztUf1jI50kVys1lq1FqcZQiziHHriAEWbnrYn6Fgq13g
\nnIws7Lw9Utotgj54tPCwJ8gEk0w4eT04tnZmPTTdW1mVhTdEjvRxEx8fotzthJuVisP
```

```
\nmFCxBPWJZ8IP9t2C/4Zi1PuqXI/8YZx8LPIcQuCRxeLURIgQrbp7
\n-----END CERTIFICATE-----\n"
```

4. Remove the '\n' character from the certificate (from BEGIN to END CERTIFICATE) and store the certificate in a file.
5. Import the certificate into the keystore that was created earlier:

```
keytool -importcert -trustcacerts -alias mydomainnew -file <new-certificate>
-keystore mydomain2.jks -storepass <keystore-password>
```

After rotating the client key, use the new certificate and private key to create locked vaults. On Transparent cloud tiering, update the cloud account by using the **mmccloudgateway account update** command.

Rotating Client key or revoking old certificate by using the automation script

- a. Run **mcstore_lockedvaultrotateclientKey.sh** <keystorenewaliasname> <keystoreoldaliasname> <keyStorePath> <COSManagerIP> <expirationDays> <COSCACertFile>, where the first 4 parameters are mandatory and the last two parameters (<expirationDays> and <COSCACertFile>) are optional.

If the expiration date (expirationDays) is not specified, then the command will take the default expiration time, which is 365 days.

If the IBM Cloud Object Storage CA certificate (COSCACertFile) is not specified, then the CA file will be downloaded from the IBM Cloud Object Storage Manager.

- b. For the description of the parameters, see the **mmccloudgateway** command.

For example, run this command:

```
./mcstore_lockedvaultrotateclientkey.sh testnew5 test /root/svt/test.ssl/test.jks 9.10.0.10
```

The system displays output similar to this:

```
Enter KeyStore Password:
Note: Before rotating the client key and certificate take a backup of old Key Store
Validating the inputs and the configuration....
COS Manager is reachable. Proceeding with Configuration...

Transparent Cloud Tiering Server RPM already installed. Proceeding with Configuration...

Python libraries are already installed. Proceeding with Configuration...

CURL already installed. Proceeding with Configuration...
Certificate stored in file </root/svt/test.ssl/test_new.crt>
MAC verified OK
writing RSA key
What is your first and last name?
[Unknown]: demo
What is the name of your organizational unit?
[Unknown]: demo
What is the name of your organization?
[Unknown]: demo
What is the name of your City or Locality?
[Unknown]: demo
What is the name of your State or Province?
[Unknown]: demo
What is the two-letter country code for this unit?
[Unknown]: IN
Is CN=demo, OU=demo, O=demo, L=demo, ST=demo, C=IN correct? (type "yes" or "no")
[no]: yes

Generating a new CSR....
Downloading COS CA Certificate....
Sending CSR to CleverSafe to be signed.....
% Total    % Received % Xferd  Average Speed   Time     Time      Time Current
          Dload Upload Total Spent   Left Speed
100  2992  100  1777  100  1215  5758  3937 --:-- --:-- --:-- 5769
Retrieving Certificate from Response.....
Importing New Client Certificate to Keystore.....
Certificate reply was installed in keystore
```

```
IMPORTANT: /root/svt/test.ssl contains private key, keystore and private certificate.  
You must keep a back up of /root/svt/test.ssl.  
Rotate Client Key Completed Successfully.  
Note: Please use mmccloudgateway update account command to import new certificate and private  
key in to TCT.  
New Alias Name is : testnew5
```

Updating Transparent cloud tiering with a new private key and certificate

This topic describes how to update Transparent cloud tiering with a new key and certificate.

1. Update the cloud account with the new private key and the certificate by issuing the following command:

```
mmccloudgateway account update --cloud-nodeclass tct --account-name mycloud  
--src-keystore-path /root/mydomain.jks --src-keystore-alias-name mydomainnew --src-keystore-  
type jks  
--src-keystore-pwd-file /root/pwd
```

2. For more information, see the **mmccloudgateway account update** command.

For example, to update the cloud account (node class tct, cloud name mycloud) with new key and certificate, issue the following command:

```
mmccloudgateway account update --cloud-nodeclass tct --cloud-name mycloud --src-keystore-path  
/root/demold/worm\*\*.ssl\xyz%\*\*.jks --src-keystore-alias-name wormnew --src-keystore-type jks  
--src-keystore-pwd-file /root/pwd
```

The system displays output similar to this:

```
mmccloudgateway: Sending the command to the first successful node starting with c350f3u30  
mmccloudgateway: This may take a while...
```

Note: Ensure that you have a backup of the Source Key Store used to import the private key and certificates. Transparent Cloud Tiering removes the private key and certificate from the trust store if the account delete command is run.

```
mmccloudgateway: Command completed successfully on c350f3u30.  
mmccloudgateway: You can now delete the password file '/root/pwd'  
mmccloudgateway: Command completed.
```

Chapter 8. Configuring IBM Power Systems for IBM Storage Scale

The following information provides guidelines on tuning IBM Power Systems for IBM Storage Scale.

To get the best results, make sure that your system is correctly configured and has the correct firmware for the server, adapters and operating system version.

Operating system considerations

IBM Storage Scale can run on IBM Power System with the Linux distribution and the kernel versions that are mentioned in the *Functional Support Matrices* tables in the [IBM Storage Scale FAQ in IBM Documentation](#).

Note: IBM has tested these guidelines on updated kernel levels that are included the relevant security updates from `rhn.redhat.com`. The minimum level suggested for IBM Power nodes that run IBM Storage Scale is RHEL 7.1.

Required packages

Ensure that all executable files and configuration files are installed specific to Linux distribution to tune the system and to help it function.

For example, on RHEL the following packages must be installed on all running partitions of IBM Storage Scale:

- `tuned-utils.noarch`
- `tuned.noarch`
- `numactl`
- `powerpc-utils`

Firmware considerations

You should upgrade your IBM Power Systems firmware to the updated version to ensure optimal performance of IBM Storage Scale continually. Obtain IBM Power Systems firmware updates from the [IBM Fix Central](#) site.

For more information, see [IBM Developer tutorial](#).

The firmware of other components in the environment, in addition to the IBM Power Servers, must be updated regularly. For example, firmware updates must be applied to network switches and network adapters that are used in an IBM Storage Scale environment. Typically, firmware updates for Mellanox adapters are obtained through the `MLNX_OFED` packages (as of May 2020, the updated level is **MLNX_OFED_LINUX-4.7-3.0.2.0**).

Tuning the operating system

Describes the operating system tuning procedure that is required for IBM Storage Scale cluster nodes.

Tune each node of the IBM Storage Scale cluster by following the procedure:

Note: Some tuning decisions require that you make choices that balance the benefits of optimal performance against the cost of potential additional energy usage. The best performance depends on the applications that are run. The configurations that are shown have demonstrated good results in IBM lab testing. However, each configuration option you choose must be based on your own requirements.

1. Install the required packages to run the tuning commands. For example, on an RHEL system, run the following command to install the required software:

```
yum -y install tuned-utils tuned numactl powerpc-utils
```

2. Set throughput performance governor by running the following command:

```
tuned-adm profile throughput-performance
```

3. Set the CPU tuning by running the following commands:

```
cpupower idle-set -e 0
```

```
cpupower idle-set -D 1
```

You can run the following command to view the CPU settings:

```
cpupower idle-info
```

Note: You can reset the default values by using the following command:

```
cpupower idle-set -E
```

4. Set the value of Simultaneous multithreading (SMT) to 2 by running the following command:

```
ppc64_cpu -smt=2
```

Logical partitioning (LPAR) hardware allocations for NUMA-based Power servers

Describes the hardware allocations that are required to tune all the components from the Logical Partition (LPAR) level up to IBM Storage Scale to achieve the best performance.

Memory and CPU provisioning for LPAR

If an IBM Power server is running with more than one LPAR, then you need to validate the way memory and CPU are provisioned to each LPAR. It is a good idea to ensure that the processes on the operating system are dispatched to a CPU on the same NUMA node to which the memory is assigned. This takes advantage of memory locality, and ensures that workloads are equally distributed across NUMA nodes. The goal is to minimize the number of NUMA nodes that an LPAR is assigned to. Determine the amount of memory available for LPARs in a NUMA node by allocating around 5-10% of each node for the server hypervisor.

If the required memory for an LPAR and the amount that is required by the hypervisor is greater than what is available in a single NUMA node, then the LPAR needs to allocate memory from multiple NUMA nodes. Allocating from multiple NUMA nodes, instead of a single NUMA node, allows for the memory to be successfully assigned to the LPAR. However, there can be an associated performance cost because non-local memory accesses will have higher memory latencies. After the LPAR boots up, you can check whether all the memory is assigned from one partition with the **numactl -H**.

The Dynamic Platform Optimizer (DPO), automatically optimizes processor and memory affinity. It is run from the HMC command line. The dynamic nature of the DPO allows it to be run while the partitions are active. For more information, see [Dynamic Platform Optimizer in IBM POWER8® Documentation](#).

For a new installation, while you are configuring the system to run for the first time, run DPO while the partitions are shut down. This process ensures that allocation can be done quickly without migrating active pointers and data. For more information, see [“Running Dynamic Platform Optimizer \(DPO\) to optimize an LPAR” on page 123](#).

Sample Scenario for Power server tuning

A sample scenario can be cited where you configure 4 LPARs on an IBM Power server with 512 GB of memory and 20 processors (across two processor modules or NUMA nodes). The partition profiles for

each LPAR must be set to Dedicated processor mode, in this case, with five processors for each LPAR. For memory, the partition profiles for each LPAR must be set with 115 GB desired and 120 GB max memory. This memory allocation reserves about 10% of the memory for the server hypervisor use. Configuring the LPARs, by using this process, allows the DPO to assign each LPAR to a dedicated NUMA node, thus providing the best performance for each LPAR. While running the DPO, depending on hardware and hypervisor restrictions, the amount of memory can be refined to reduce the amount of wasted memory.

If adapters must also be virtualized, a VIO server may need to be configured. Two VIO servers help to ensure high availability. In such a scenario, the memory allocated to each LPAR needs to be reduced to allow the VIO servers to run as well.

Running Dynamic Platform Optimizer (DPO) to optimize an LPAR

Describes the procedure that you need to follow for running DPO to optimize an LPAR for IBM Storage Scale.

To run DPO to optimize LPAR for IBM Storage Scale, you need to ensure that the best memory affinity is made available to the partitions. Availability of memory is relevant when there are more than one LPAR per system.

Note: The following procedure is not required if you have a single system that has all processors and memory that is allocated to a single partition.

1. Shut down the operating system to display the LPAR status as Not Activated.
2. Change the partition profile for IBM Storage Scale. For more information, see “[Logical partitioning \(LPAR\) hardware allocations for NUMA-based Power servers](#)” on page 122.
3. Activate the changed profile and load the operating system by pointing to SMS.

This moves the memory and processor configuration, you created from the profile that is resident in the HMC, to the Hypervisor without the need to load the operating system.

4. Shut down the partitions again.
5. Run the Dynamic Platform Optimizer (DPO) for the Server after all the partitions on it are correctly configured by using the following command in the HMC command line:

```
optmem -m <managed server> -t affinity -o start
```

6. Run the following command from the HMC command line to monitor the progress of the DPO process:

```
lsmemopt -m <managed server>
```

7. Activate the partitions by using the current configuration when the DPO process is completed and the partitions are fully contained (both processor and memory) to their own processor module.

Configuring INT_LOG_MAX_PAYLOAD_SIZE Parameter

Describes the configuration of the Mellanox device firmware parameter **INT_LOG_MAX_PAYLOAD_SIZE**, on IBM Storage Scale NSD Servers, Protocol nodes, and clients.

Run the command shown to view the current settings of **INT_LOG_MAX_PAYLOAD_SIZE** on all Mellanox devices on the system.

```
for i in `ls /sys/class/infiniband/`; do mlxconfig -d $i -e q INT_LOG_MAX_PAYLOAD_SIZE; done
```

Note: You need to install the mlxconfig tool. For more information, see [Mellanox Firmware Tools User Manual](#).

The details of the command output are displayed in the tables shown.

Table 7. Device details for Device 1

Device Type	Name	Description	Device
ConnectX5	00WT174_Ax	PCIe4 2-port VPI EDR IB (100 Gb/s) and 100 GbE Adapter x16	mlx5_0

Table 8. Default Configurations for INT_LOG_MAX_PAYLOAD_SIZE

Configurations			
INT_LOG_MAX_PAYLOAD_SIZE	AUTOMATIC(0)	AUTOMATIC(0)	AUTOMATIC(0)

INT_LOG_MAX_PAYLOAD_SIZE Settings

Setting 1: _4KB: 4 KB burst length (set by mlxconfig [...] INT_LOG_MAX_PAYLOAD_SIZE=12)

This is the non-default setting that is recommended for all IBM Storage Scale NSD servers and protocol nodes. It ensures the optimal performance, particularly for large I/O streaming use cases when RDMA is enabled.

Run the following command to configure _4KB(12) as the value for the **INT_LOG_MAX_PAYLOAD_SIZE** parameter of all Mellanox devices:

```
for i in `ls /sys/class/infiniband/`; do mlxconfig -d $i -e s INT_LOG_MAX_PAYLOAD_SIZE=12 ; done
```

The values that are displayed in the command output are shown in the following table.

Table 9. Recommended Configurations for INT_LOG_MAX_PAYLOAD_SIZE Values

Configuration	Default	Current	Next Boot
*INT_LOG_MAX_PAYLOAD_SIZE	AUTOMATIC(0)	_4KB(12)	_4KB(12)

The * shows parameters with the next value that is different from the default or the current value.

Setting 2: AUTOMATIC: Default (set by mlxconfig [...] INT_LOG_MAX_PAYLOAD_SIZE=0)

The default setting is applicable to all IBM Power nodes that are **not** configured as an IBM Storage Scale server or protocol node. For example, compute nodes that are only clients of an IBM Storage Scale file system.

Run the following command to apply the default setting:

```
for i in `ls /sys/class/infiniband/`; do mlxconfig -d $i -e s INT_LOG_MAX_PAYLOAD_SIZE=0; done
```

Note: In both scenarios, after you run the commands, you need to restart your system for the changes to be effective.

Chapter 9. Configuring file audit logging

The following topics describe various ways to configure file audit logging in IBM Storage Scale.

Enabling file audit logging on a file system

Use this information to enable file audit logging on a file system in IBM Storage Scale.

Important:

- Only one type of file audit logging can be enabled per file system. You can either enable file system auditing, fileset auditing, or skip fileset auditing.

1. To enable a file system for file audit logging, run the **mmaudit** command.

```
mmaudit Device enable
```

2. To update the events being audited, filesets to audit or filesets to skip, run the mmaudit command.

```
mmaudit Device update --events OPEN,CLOSE
```

For more information, see the **mmaudit** command in the *IBM Storage Scale: Command and Programming Reference Guide*.

Note:

- If the Object protocol is enabled on the file system that contains the file audit logging fileset, ensure that additional inodes are defined for it before you enable file audit logging.
- In an ESS environment, file audit logging must be enabled manually on non-EMS and ESS I/O nodes.
- For more information about validating that a node is getting events after file audit logging is enabled, see *Monitoring the file audit logging fileset for events* in the *IBM Storage Scale: Problem Determination Guide*.

Note: Enabling file audit logging is audited and recorded by syslog. For more information, see *Audit messages for cluster configuration changes* in the *IBM Storage Scale: Problem Determination Guide*.

Disabling file audit logging on a file system

Use this information to disable file audit logging on a file system in IBM Storage Scale.

To disable file audit logging on a file system, issue the **mmaudit** command.

```
mmaudit Device disable
```

Note:

- The audit log fileset is not deleted during disablement.
- The **disable** command works the same way for all audit types. Whether file system, fileset, or skip fileset audit is configured, the **mmaudit <fs> disable** command disables all types.

For more information, see the **mmaudit** command in the *IBM Storage Scale: Command and Programming Reference Guide*.

Note: Disabling file audit logging is audited and recorded by syslog. For more information, see *Audit messages for cluster configuration changes* in the *IBM Storage Scale: Problem Determination Guide*.

Enabling or skipping filesets with file audit logging

With fileset file audit logging, you can specify a list of filesets to apply file audit logging to or to skip it from within a file system.

To enable file audit logging so that it is applied to a specific list of filesets only, run a command similar to the following example:

```
mmaudit Device enable --filesets {Fileset[,Fileset...]|ListFilePath|}
```

To enable file audit logging so that it is applied to an entire file system except a specific list of filesets, run a command similar to the following example:

```
mmaudit Device enable --skip-filesets {Fileset[,Fileset...]|ListFilePath|}
```

Note:

- For an example, see the **mmaudit** command in the *IBM Storage Scale: Command and Programming Reference Guide*.
- For more information about validating that a node is getting events after file audit logging is enabled, see *Monitoring the file audit logging fileset for events* in the *IBM Storage Scale: Problem Determination Guide*.

Important:

- Filesets can be independent or dependent.
- The **--enable-filesets** option cannot be used in combination with the **--disable-filesets** option. There can only be one audit type updated at a time.
- The root fileset can be specified as input.
- The **--skip-filesets** option cannot be used in combination with the **--filesets** option. There can only be one audit type per file system: file system, fileset, or skip fileset.
- None of the listed filesets can be .msgq, the audit fileset, the **cesSharedRoot**, or the default Object fileset.
- There is a limit of 20 filesets for file system prior to 5.1.3 (27.0 file system version).
- All filesets in the lists of the **--filesets**, **--skip-filesets**, **--enable-filesets** or **--disable-filesets** option must be linked in the file system that is specified by **mmaudit Device enable** or **update**.
- Events are not generated for nested dependent or independent filesets under an independent fileset that is specified by **--filesets** or **--enable-filesets** option.
- Events are generated for nested dependent or independent filesets under an independent fileset that is specified by **--skip-filesets** or **--disable-filesets** option.

Actions that the **mmaudit** command takes to enable file audit logging

Describes the actions that are taken when the **mmaudit** command enables file audit logging.

Note:

- If any of the following steps fail, then the entire enablement of file audit logging for the file system fails.
 - Any updates to the configuration or filesets are rolled back so that the cluster is not left in a state where a file system is partially enabled for file audit logging.
1. Checks that the file system is mounted locally. The file system is the file system where the file audit logging fileset is located that contains the audit log files.
 2. Updates the audit configuration with the file audit logging configuration information, which includes the file system that is being audited and the audit log fileset name among other attributes.

- If it is needed, it creates the audit log fileset on the device that is specified in the configuration. By default, the fileset is created in IAM mode non compliant. If the **--compliant** flag is specified when file audit logging is being enabled, then the fileset is created in IAM mode compliant.

Note: For more information, see *The file audit logging fileset* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

- Creates the policy partitions that are used to receive lightweight events and block lightweight events from certain filessets and paths:
 - Creates the policy partition that is used to receive lightweight events for the file system device that is being audited.
 - If needed, creates policy partitions to skip file system operations in the file audit logging fileset.
 - If needed, creates global policy partitions to skip known paths (such as the CES shared root) where lightweight events are not wanted.
- Changes the file system configuration so that the **mmlsfs** command with the **--file-audit-log** option shows as yes.

To verify the settings of one or more file systems that are enabled for file audit logging, run the **mmaudit all list** command. This command displays the configuration information for all file systems that are configured for file audit logging. Running the **mmaudit all list** command displays output similar to the following example:

Audit Device	Cluster ID	Audit Fileset Name	Retention (Days)	Audit Type (Possible Filesets)
fs0	11430652110915196903	john1	25	FILESET dep1,dep2,ind1,ind2
fs1	11430652110915196903	john2	75	SKIPFILESET dep1,dep2,ind1,ind2
fs2	11430652110915196903	john3	25	FSYS

Actions that the **mmaudit** command takes to disable file audit logging

This topic describes how the **mmaudit** command disables file audit logging.

- Removes the associated policy partitions that were used to receive lightweight events and block lightweight events from certain filessets and paths:
 - If they exist, removes the global policy partitions that were used to skip paths (such as the CES shared root) where lightweight events are not wanted.
 - If it is needed, removes the policy partition that was used to skip file system operations in the file audit logging fileset.
 - Removes the policy partition that was used to receive lightweight events for the file system device that was being audited.
- Updates the audit configuration to remove the file audit logging configuration information that is associated with the file system that was audited.
- Makes the change to the file system configuration, so that the **mmlsfs** command with the **--file-audit-log** option shows as no.

Enabling and disabling file audit logging using the GUI

For more information about enabling and disabling file audit logging using the GUI, see “[Managing file audit logging](#)” on page 264.

Viewing file systems that have file audit logging enabled with the GUI

You can use the **Files > File Systems** page in the IBM Storage Scale management GUI to monitor whether file audit logging is enabled for file systems.

The **File Audit** column in the file systems table displays which file systems are file audit logging enabled. The **File Audit** column is hidden by default. To see whether file audit logging is enabled, perform the following steps:

1. Go to **Files > File Systems** in the management GUI.
2. Select **Customize Columns** from the **Actions** menu.
3. Select **File Audit**. The **File Audit** column is visible now.

Enabling file audit logging on an owning cluster for a file system that is remotely mounted

Use this information to enable file audit logging on an owning cluster for a file system that is remotely mounted.

Perform the following steps to enable file audit logging for a remotely mounted file system:

1. Make sure that both the accessing and owning clusters are on IBM Storage Scale 5.0.2 minimum release level.
2. Make sure that the file systems that are going to be remotely mounted are at IBM Storage Scale 5.0.2 or higher.
3. Follow the instructions in *Accessing a remote GPFS file* in the *IBM Storage Scale: Administration Guide*.
4. Validate that the accessing cluster has the required packages installed by referring to *Requirements and limitations of file audit logging* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
5. If not already enabled, enable file audit logging in the owning cluster by following the instructions in Chapter 9, “Configuring file audit logging,” on page 125.
6. At this point, file audit logging should be logging all file system activity from the accessing cluster nodes that have fulfilled the previous steps.
7. The file audit logs will be owned and located on the owning cluster. Run **mmaudit <device> list** on the owning cluster for details.

Note: The file audit logging producers on the accessing cluster will log debug messages to the local `/var/adm/ras/mmaudit.log` file. But the overall file audit logging status and logging can only be obtained from the owning cluster. For more information, see *File audit logging issues* in the *IBM Storage Scale: Problem Determination Guide*.

Chapter 10. Configuring clustered watch folder

After clustered watch folder is installed, it can be enabled, disabled, or configured with an external Kafka sink.

Enabling a clustered watch

Use this information to enable a clustered watch.

To enable a clustered watch on a file system, run the **mmwatch** command:

```
mmwatch <Device> enable --event-handler kafkasink --sink-brokers "BrokerIP:Port" --sink-topic  
"Topic"
```

To enable a clustered watch on a fileset, run the **mmwatch** command:

```
mmwatch <Device> enable --fileset <fsetname> --event-handler kafkasink --sink-brokers  
"BrokerIP:Port" --sink-topic "Topic"
```

For more information, see “[Actions that the mmwatch command takes to enable a clustered watch](#)” on page 129. For more information about using the **--sink-auth-config** flag, see the *Interaction between clustered watch folder and the external Kafka sink topic in IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Disabling a clustered watch

Use this information to disable a clustered watch.

To disable a clustered watch on a file system or a fileset, run the **mmwatch** command:

Note: The watch ID for a specific watch must be passed into the **mmwatch** command. The watch ID can be found by running the **mmwatch all list** command.

```
mmwatch <Device> disable --watch-id <WatchID>
```

To disable all watches, run the **mmwatch** command:

```
mmwatch <Device> disable --watch-id all
```

For more information about what happens when a clustered watch is disabled, see “[Actions that the mmwatch command takes to disable a clustered watch](#)” on page 130.

Configuration of an external Kafka sink in the IBM Storage Scale cluster

Use this information to set up an external Kafka sink on IBM Storage Scale nodes.

Follow the instructions in the official Apache Kafka quick start guide: <https://kafka.apache.org/quickstart>. Kafka can be installed on any IBM Storage Scale cluster/node as long as there is not a message queue enabled.

Actions that the mmwatch command takes to enable a clustered watch

This topic describes how the **mmwatch** command enables a clustered watch in IBM Storage Scale.

Note: If any of these steps fail, all of the previous steps are returned to the pre-enablement state.

1. The **mmwatch** command verifies that all of the required parameters are present either through the command line or the input file.
2. The **mmwatch** command verifies that the file system that is associated with the clustered watch is mounted on the local node.
3. The **mmwatch** command converts the event list to a hex bit mask.
4. The **mmwatch** command creates a topic with a name in the following format:

```
SpectrumScale_WF_C_<ClusterID>_<WatchType>_<WatchID>_CLW_<Device>
```

5. The **mmwatch** command creates and uploads the CCR file that corresponds to the requested watch. The name of the file is in the following format:

```
_SpectrumScale_WF_C_<ClusterID>_<WatchType>_<WatchID>_CLW_<Device>
```

6. If it is necessary (for example if the sink authentication type is CERT), the **mmwatch** command pushes the certificate files to all of the producer nodes.

Note: For more information, see *Interaction between clustered watch folder and the external Kafka sink* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

7. The **mmwatch** command creates the policy partition for the newly created watch.

Actions that the **mmwatch** command takes to disable a clustered watch

This topic describes how the **mmwatch** command disables a clustered watch in IBM Storage Scale.

The **mmwatch** command takes the following actions to disable a clustered watch:

1. Checks if the file system that is associated with the clustered watch is mounted on the current node and cleans up the configuration fileset. The disable will still occur if the file system is not mounted but the configuration fileset will remain.
2. If it was the last enabled clustered watch, deletes the configuration fileset skip partition.
3. Removes the active clustered watch policies.
4. Removes CCR-based configuration and fileset-based configuration.

Chapter 11. Configuring the **cloudkit**

To communicate with a specific cloud, the **cloudkit** must be configured with cloud credentials. The following topics describe methods to configure **cloudkit**.

Configuring your AWS cloud account

Configure **cloudkit** with your Amazon Web Services (AWS) credentials to perform cloud provisioning operations.

Prerequisites

The following prerequisites must be met before you run the **cloudkit configure** command:

- Create an account in [AWS](#), if you do not already have one.
- Create [access keys](#). Access keys consist of two parts, an access key ID and a secret access key.

Configuration of **cloudkit**

To configure **cloudkit**, issue the following command:

```
./cloudkit configure
```

Note: If the **cloudkit** is installed on cloudVM, **configure** is not required.

AWS example:

```
# ./cloudkit configure
I: Logging at /root/scale-cloudkit/logs/cloudkit-10-9-2023_8-28-13.log
? Cloud platform name: AWS
? AWS Access Key ID: [? for help] *****
? AWS Secret Access Key: [? for help] *****
I: Local machine has been configured to use your 'AWS' account.
```

Configuring your GCP cloud account

Configure **cloudkit** with your Google Cloud Platform (GCP) credentials to perform cloud provisioning operations.

Prerequisites

The following prerequisites must be met before you run the **cloudkit configure** command:

- A service account with sufficient privileges and quota to provision all required resources along with the credentials to access the GCP API. To create a service account from the GCP console, refer to GCP documentation, [Create a service account](#).
- Service account credentials JSON file. To generate and download the JSON file, refer to GCP documentation, [Create credentials for a service account](#).

Configuration of **cloudkit**

To configure **cloudkit**, issue the following command:

```
./cloudkit configure
```

GCP example:

```
# ./cloudkit configure
I: Logging at /root/scale-cloudkit/logs/cloudkit-10-9-2023_8-24-23.log
```

```
? Cloud platform name: GCP
? GCP Project ID: projectname-349401
? GCP Service user credential json path: /home/metadata/projectname-349401-6d757b56bd62.json
I: Local machine has been configured to use your 'GCP' account.
```

Configuring your Azure cloud account

Configure **cloudkit** with your Azure credentials to perform cloud provisioning operations.

Prerequisites

The following prerequisites must be met before you run the **cloudkit configure** command:

- A service account (service principle) with sufficient privileges and quota to provision all required resources along with the credentials to access the Azure API. To create a service account from the Azure console, refer to Azure documentation, [Create an Azure service principal with Azure CLI](#).

Configuration of **cloudkit**

To configure **cloudkit**, issue the following command:

```
./cloudkit configure
```

Azure example:

```
# ./cloudkit configure
I: Logging at /root/scale-cloudkit/logs/cloudkit-16-7-2024_9-12-41.log
? Cloud platform name: Azure
? Client ID (or Application ID): xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
? Client Secret (or Client Key or Client Secret Key): [? for help]
*****
? Azure Tenant ID (or Directory ID or Directory Tenant ID): xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
? Azure Subscription ID: xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
I: Local machine has been configured to use your 'Azure' account.
```

Chapter 12. Configuring Active File Management

For Active File Management (AFM) configuration, all AFM parameters are listed with all important information in different tables. Also, you can configure AFM by following the configuration-related steps.

Configuration parameters for AFM, AFM-DR, and AFM to cloud object storage

You can configure AFM, AFM-DR and AFM to COS by using configuration parameters with their defined properties, such as, default values, valid values, unit, and so on.

The parameters for parallel data transfer can be used at the cache cluster. Some parameters do not take effect until the GPFS daemons on the AFM gateway nodes are shut down and restarted. The following table lists the AFM and AFM-DR configuration parameters.

Table 10. AFM, AFM-DR, and AFM to cloud object storage configuration parameters

Parameter ¹	Default value ²	Valid value ³	Unit ⁴	Application ⁵	Parallel transfer ⁶	AFM file cache ⁷	AFM object cache ⁸	AFM-DR ⁹
afmAsyncDelay Indicates the time when the requests start flushing to the home or the secondary cluster. This asynchronous delay is helpful for write-intensive applications that write to the same set of files. Because of this delay multiple writes to the home or the secondary cluster are replaced with a single write, which contains the latest data. However, if the parameter value is set high, the data updates on a remote cluster cause inconsistency.	15	1 - 2147483647	Second	Cluster, fileset		SW, IW		Primary
afmAsyncOpWaitTimeout Specifies the time when AFM waits for completion of any inflight asynchronous operation that is synchronizing with the home or the primary cluster. AFM cancels the asynchronous operation and synchronizes again after the home or the primary cluster is available.	300	5 - 2147483647	Second	Cluster		IW, RO, SW		Primary
afmDirLookupRefreshInterval Defines the frequency of revalidation that is triggered by a look-up operation such as ls or stat on a directory from the cache cluster. AFM sends a message to the home cluster to find out whether the metadata of that directory is modified since it was last revalidated. If so, the latest metadata information at the home cluster is reflected on the cache cluster.	60	0 - 2147483647	Second	Cluster, fileset		RO, LU, IW		

Table 10. AFM, AFM-DR, and AFM to cloud object storage configuration parameters (continued)

Parameter ¹	Default value ²	Valid value ³	Unit ⁴	Application ⁵	Parallel transfer ⁶	AFM file cache ⁷	AFM object cache ⁸	AFM-DR ⁹
afmDirOpenRefreshInterval Defines the frequency of revalidations that are triggered by the read and update operations on a directory from the cache cluster. AFM sends a message to the home cluster to find whether the metadata of that directory is modified since it was last revalidated. Open requests on files or subdirectories on that directory are served from the cache fileset until the afmDirOpenRefreshInterval expires after which the open requests are sent to the home cluster.	60	0 - 2147483647	Second	Cluster, fileset		RO, LU, IW		
afmObjectDirectoryObj AFM to cloud object storage supports directory objects. All directories with and without objects can now be synchronized to the cloud object storage. You can now set extended attributes on directories with the directory object support. To enable this afmObjectDirectoryObj parameter, stop the fileset, enable this parameter and start the fileset.	no	yes no	Boolean	fileset	NA	NA	RO, SW, IW and LU	NA
afmDisconnectTimeout Defines the interval until which the AFM MDS or the primary cluster waits after it detects that the home or the secondary cluster is inaccessible before declaring the outage by moving the cache or the primary cluster state to Disconnected.	60	0 - 2147483647, disable	Second	Cluster		RO, SW, IW, LU		Primary
afmEnableAutoEviction Indicates whether automatic eviction is triggered on a fileset.	no	yes no	Boolean	Fileset		RO, SW, IW, LU		
afmEnableNFSSec If enabled on the cache or the primary cluster, exported paths from the home or the secondary cluster with Kerberos-enabled security levels like sys, krb5, krb5i, krb5p are mounted on the cache or the primary cluster in the increasing order of security level - sys, krb5, krb5i, krb5p. For example, the security level of exported path is krb5i then on the cache or the primary cluster, AFM tries to mount with level sys, followed by krb5, and finally mounts with the security level krb5i. If disabled on the cache or the primary cluster, exported paths from the home or the primary cluster are mounted with security level sys on the cache or primary cluster. You must configure KDC clients on all the gateway nodes on the cache or the primary cluster before you enable this parameter.	no	yes no	Boolean	Cluster		IW, LU, RO, SW		Primary

Table 10. AFM, AFM-DR, and AFM to cloud object storage configuration parameters (continued)

Parameter ¹	Default value ²	Valid value ³	Unit ⁴	Application ⁵	Parallel transfer ⁶	AFM file cache ⁷	AFM object cache ⁸	AFM-DR ⁹
afmExpirationTimeout Is used with afmDisconnectTimeout to control the duration of a network outage between the cache and home clusters before the data in the cache expires and becomes unavailable until a home reconnection occurs.	disable	0 - 2147483647 , disable	Second	Cluster, filesset		RO		
afmEvictRange Is used with the mmafmctl command, --range option. When afmEvictRange option is set on the filesset, auto-eviction and manual eviction (without range) evicts all the data blocks except first and last data blocks.	no	yes no	Boolean	Fileset		RO, IW, SW, LU	RO, IW, SW, LU	
afmFastCreate Enable at the AFM cache and AFM-DR primary filesset level. AFM sends RPC to the gateway node for each update that is happening on the filesset. If the workload mostly involves new files creation, this parameter reduces the RPC exchanges between the application and the gateway node, improves the application performance, and minimizes the memory queue requirement at the gateway node.	no	yes no	Boolean	Fileset		IW, SW		Primary
afmFastLookup Modifies the behavior that improves AFM performance when requests come for operations, such as lookup, open, readdir, or getattr, when AFM must fetch or read configuration of the filesset. After you enable this parameter, AFM can store and access the AFM filesset configuration information from the last successful refresh operation directly from the cache. This optimization improves the performance of lookup, open or getattr calls for AFM. If this parameter is not enabled, AFM must use the read-only locking mutex for the filesset configuration, which sometimes might lead to a locking contention. Hence, slowness is observed during lookup and readdir operations.	disable	yes no	Boolean	Fileset				
afmFileLookupRefreshInterval Defines the frequency of revalidation that is triggered by a look-up operation on a file such as ls or stat, from the cache cluster. AFM sends a message to the home cluster to determine that the metadata of the file is modified since it was last revalidated. If so, the latest metadata information at home is reflected on the cache cluster.	30	0 - 2147483647	Second	Cluster, filesset		IW, LU, RO		

Table 10. AFM, AFM-DR, and AFM to cloud object storage configuration parameters (continued)

Parameter ¹	Default value ²	Valid value ³	Unit ⁴	Application ⁵	Parallel transfer ⁶	AFM file cache ⁷	AFM object cache ⁸	AFM-DR ⁹
afmFileOpenRefreshInterval Defines the frequency of revalidations that are triggered by the read and write operations on a file from the cache cluster. AFM sends a message to the home cluster to determine that the metadata of the file is modified since it was last revalidated. Open requests on the file are served from the cache fileset until the afmFileOpenRefreshInterval expires after which the open requests are sent to the home cluster.	30	0 - 2147483647	Second	Cluster, fileset		IW, LU, RO		
afmGateway=GatewayNode Specifies the user-defined gateway node for an AFM or AFM DR fileset. When this parameter is set, it takes preference over the default gateway node that is assigned by AFM by using internal hashing algorithm. If the specified gateway node is not available, AFM internally assigns a gateway node from the available list by using the hashing algorithm. This parameter value can be set to afmGateway = all for the AFM RO mode and the AFM LU mode during the file system-level migration. When this value is set, it improves the performance of data migration. This parameter with all value enforces AFM to migrate data by distributing migration task that is pertaining to inodes or files at the gateway node. afmHashVersion=5 must be set at the cluster level by using the nmchconfig afmHashVersion=5 command.	None	Gateway node name	String	Fileset		IW, LU, RO, SW (Value 'all' can be used for LU and RO.)		Primary
afmHardMemThreshold Sets the maximum memory that AFM can use on each gateway node for handling queues. After this limit is reached, queues might not be handled on this gateway node due to lack of sufficient memory. Filesets belonging to this gateway node might go to a 'Dropped' state, depending on the activity. Exceeding the limit can occur if the cache cluster is disconnected for an extended time or if the connection with the home cluster has low bandwidth and therefore a lot of pending requests are accumulated in the queue. After the value of this parameter is changed, a gateway node daemon recycle is required for the new value to take effect.	5 GiB		Bytes	Cluster	Yes			Primary

Table 10. AFM, AFM-DR, and AFM to cloud object storage configuration parameters (continued)

Parameter ¹	Default value ²	Valid value ³	Unit ⁴	Application ⁵	Parallel transfer ⁶	AFM file cache ⁷	AFM object cache ⁸	AFM-DR ⁹
afmHashVersion Specifies the gateway node hashing algorithm that minimizes the impact of gateway nodes joining or leaving the active cluster by running as few recoveries as possible and balance mapping of AFM or AFM-DR filesets across the gateway node. Valid values are 1, 2, 4, and 5. Default value is 2. You can specify the value by using the mmchconfig command. For example, fit5, run the mmchconfig afmHashVersion=5 command.	2	Number	1 2 4 5	Cluster		IW, LU, RO, SW		Primary, Secondary
afmHomeSnapshotName This parameter can be enabled only when a home site is not an IBM Storage Scale or GPFS site. When this parameter is set, AFM detects the snapshot path on the non-IBM Storage Scale or GPFS home during data migration and prefetch cases.	.snapshot s	user defined	String	Cluster		LU, RO		
afmLookupMapSize Creates a map of a specified size on a gateway node to store values of the last lookup refresh and the readdir refresh. AFM uses these values when the AFM filesset receives any operation request, such as lookup, readdir, open, getattr. Note: Enable this parameter with the afmFastLookup parameter. This map helps AFM to determine whether re-validation of configuration is required by checking the latest cache entries populated in the map. This parameter improves the performance and prevents any locking mutex contention that might be encountered in the cluster. This parameter is not used for the following in the following cases: <ul style="list-style-type: none">• The afmRefreshOnce, afmReaddirOnce parameters are set.• The request is coming from the remote AFM configuration.	0	0 through 30						
afmMaxParallelRecoveries Specifies the number of filessets in the cluster on all file systems, on which recovery is run.	0	0 - 128	Whole number	Cluster		IW, LU, RO, SW		

Table 10. AFM, AFM-DR, and AFM to cloud object storage configuration parameters (continued)

Parameter ¹	Default value ²	Valid value ³	Unit ⁴	Application ⁵	Parallel transfer ⁶	AFM file cache ⁷	AFM object cache ⁸	AFM-DR ⁹
afmMode Specifies the mode in which the cache operates. Changing from single-writer/read-only modes to read-only/local-updates/single-writer is supported. When changing from read-only to single-writer, the read-only cache is up-to-date. When changing from single-writer to read-only, all requests from cache should have been played at home. Changing from local-updates to read-only/local-updates/single-writer is restricted. A typical dataset is set up to include a single cache cluster in single-writer mode (which generates the data) and one or more cache clusters in local-updates or read-only mode. AFM single-writer/independent-writer filesets can be converted to primary. Primary/secondary filesets cannot be converted to AFM filesets.			String	Fileset		IW, LU, RO, SW		Primary, secondary
afmMountRetryInterval Specifies the interval after which the primary gateway retries an operation on the home or the secondary cluster, in cases where the home or the secondary cluster is in an unhealthy state (see the Unmounted, Dropped states). Needs the GW node recycle or -i option for immediate effect. This parameter can be set per gateway. The updated value is visible until the gateway is active. This parameter value is applicable for all filesets that are owned by the gateway node.	300	1 - (2 GiB - 1)	Second			IW, LU, RO, SW		Primary
afmMUAutoRemove When the afmMUAutoRemove option is set to 'yes' on MU mode fileset, remove operations on the files are queued on the gateway node to delete the files at cloud object storage automatically. After this option is set to yes, it overrides -from-cache and -from-target options of mmafmcosctl command.	no	yes no	Boolean	Fileset			MU	
afmObjMUCheckFName Resets old AFM attributes on a disabled AFM fileset data and update the data with the new attributes. This option must be enabled, if any fileset is converted or promoted to an AFM MU mode fileset.	no	yes no	Boolean					
afmNFSv4 Is used with the mmchfileset command. Specifies NFS version 4 (NFSv4) that can be enabled on individual AFM and AFM DR filesets. When you enable this parameter on a specified fileset, the fileset behavior to use NFSv4 protocol version is modified. However, other filesets in the cluster can continue using other NFS version.	no	yes no	Boolean	Fileset		RO, IW, SW, LU		Primary

Table 10. AFM, AFM-DR, and AFM to cloud object storage configuration parameters (continued)

Parameter ¹	Default value ²	Valid value ³	Unit ⁴	Application ⁵	Parallel transfer ⁶	AFM file cache ⁷	AFM object cache ⁸	AFM-DR ⁹
afmNFSVersion Enables AFM NFSv4 supports. Default value is 3 for compatibility with an earlier version. Allowed values are 3, 4.1 and 4.2 for Knfs protocol, and 3 and 4.1 for the NFS protocol.	3	3, 4.1, 4.2	Number	Cluster	Yes	IW, LU, RO, SW		Primary, secondary
afmNumFlushThreads Defines the number of threads used on each gateway to synchronize updates to the home cluster. The default value is 4, which is sufficient for most installations. The current maximum value is 1024, which is too high for most installations. Ensure that you do not set this parameter to a very high value.	4	1 - 1024	Whole number	Fileset	Yes	IW, SW		Primary
afmNumReadThreads Defines the number of threads used on each participating gateway node during a parallel read. The default value of this parameter is 1. That is, one reader thread is active on every gateway node for each big read operation qualifying for splitting as per the parallel read threshold value.	1	1 - 64		Cluster, fileset	Yes	IW, LU, RO, SW		
afmNumWriteThreads Defines the number of threads used on each participating gateway node during a parallel write. The default value of this parameter is 1. That is, one writer thread is active on every gateway node for each big write operation qualifying for splitting as per the parallel write threshold value.	1	1 - 64	Whole number	Cluster, fileset	Yes	IW, SW		Primary
afmObjectFastReaddir Improves the objects download and readdir performance, when the afmObjectFastReaddir parameter value is set to 'yes' at the fileset level. Extended attributes and ACLs are not fetched from a cloud object storage when this parameter is enabled. Also, deleted objects on a cloud object storage system are not reflected immediately on a cache when this parameter is enabled. You can use this option to pull the objects into the cache to run quick analytics on the target data.	no	yes no	Boolean	Fileset		IW, LU, RO, SW	Yes	

Table 10. AFM, AFM-DR, and AFM to cloud object storage configuration parameters (continued)

Parameter ¹	Default value ²	Valid value ³	Unit ⁴	Application ⁵	Parallel transfer ⁶	AFM file cache ⁷	AFM object cache ⁸	AFM-DR ⁹
afmObjKeyExpiration Specifies COS Key Expiration timeout value (in seconds). In case, the expiration timeout is set for access keys and secret keys at the Cloud Object Server, you can set expiration at the cache side for AFM to reload the access key and secret key values into the memory after the defined timeout. Initially keys are loaded into the memory first time when AFM Fileset is accessed. After the expiration timeout is passed, AFM reload access and secret keys again into the memory and use the keys for communication purpose. You must update the access key and secret key once it is expired, before you start the next communication with server. The afmObjKeyExpiration parameter is set at the cluster level. The valid values are 0 to 2147483647. The default is 36000.	36000	0 - 2147483647	Seconds	Cluster			RO, SW, IW, LU, MU	
afmParallelMounts When this parameter is enabled, the primary gateway node of a fileset at a cache cluster attempts to mount the exported path from multiple NFS servers that are defined in the mapping. Then, this primary gateway node sends unique messages through each NFS mount to improve performance by transferring data in parallel. Before you enable this parameter, define the mapping between the primary gateway node and NFS servers by issuing the mmapmconfig command.	no	yes no	Boolean	Cluster, fileset		IW, LU, RO, SW		Primary
afmParallelReadChunkSize Defines the minimum chunk size of the read that needs to be distributed among the gateway nodes during parallel reads. A zero (0) value disables the parallel reads across multiple gateways. The parallel reads are routed through a single gateway node.	128	0 - 2147483647	Bytes	Cluster, fileset	Yes	IW, LU, RO, SW		
afmParallelReadThreshold Defines the threshold beyond which parallel reads become effective. Reads are split into chunks when the file size exceeds this threshold value. Values are in MB. The default value is 1024 MB.	1024	0 - 2147483647	MiB	Cluster, fileset	Yes	IW, LU, RO, SW		
afmParallelWriteChunkSize Defines the minimum chunk size of the write that needs to be distributed among the gateway nodes during parallel writes. A zero (0) value disables the parallel writes across multiple gateways. The parallel writes are routed through a single gateway node.	128	0 - 2147483647	Bytes	Cluster, fileset	Yes	IW, SW		Primary

Table 10. AFM, AFM-DR, and AFM to cloud object storage configuration parameters (continued)

Parameter ¹	Default value ²	Valid value ³	Unit ⁴	Application ⁵	Parallel transfer ⁶	AFM file cache ⁷	AFM object cache ⁸	AFM-DR ⁹
afmParallelWriteThreshold Defines the threshold beyond which parallel writes become effective. Writes are split into chunks when file size exceeds this threshold value. Values are in MB. The default value is 1024 MB.	1024	0 - 2147483647	MiB	Cluster, fileset	Yes	IW, SW		Primary
afmPrefetchThreshold Controls the partial file caching feature. 0 Full file prefetching after three blocks are read. 1-99 The percentage of the file size that must be cached before the entire file is pulled into the cache cluster. 100 Disables full file prefetching. Only fetches and caches data blocks that are read by the application. When all data blocks are cached, the file is marked as cached.	0	0 - 100	Whole number	Fileset		IW, LU, RO, SW		
afmPrimaryID Specifies the unique primary ID of the primary fileset for asynchronous data replication. This is used for connecting a secondary to a primary.				Fileset				Primary
afmReadDirOnce Enables AFM to perform one-time <code>readdir</code> of a directory from the home after the data migration to the cache and the application is started on the cache data. That is, the application is moved from the home to the cache and the application modifies the directory, which makes the directory dirty. When this parameter is set for a fileset, the prefetch operation is run on the fileset by using <code>--readdir-only</code> to move new or modified data from the home to the cache, even if the cache directory is dirty. After the data migration to the cache and the application is started on the cache data, this parameter synchronizes new files at the home directory to the cache for a single time.	no	yes no	Boolean	Fileset		IW, LU, RO		
afmReadSparseThreshold When a sparse file at the home cluster is read into the cache, the cache cluster maintains the sparseness, if the size of the file exceeds the afmReadSparseThreshold . If the size of a file is less than the threshold, sparseness is not maintained at the cache cluster.	128 M	0 - 2147483647	Bytes	Cluster, fileset		IW, LU, RO, SW		

Table 10. AFM, AFM-DR, and AFM to cloud object storage configuration parameters (continued)

Parameter ¹	Default value ²	Valid value ³	Unit ⁴	Application ⁵	Parallel transfer ⁶	AFM file cache ⁷	AFM object cache ⁸	AFM-DR ⁹
afmRecoveryDir Is used with the mmchconfig , mmchfileset , and mmcrfileset commands. Specifies a custom path for an AFM fileset. This custom path is used for the recovery or resynchronization instead of the AFM default recovery path. By enabling this parameter, you can avoid issues such as the /var/mmfs/afm partition full or no-space. These issues might occurred during the recovery of AFM filesets that have huge data.	no	Valid file path	String	Cluster		RO, IW, SW, LU	RO, IW, SW, LU	Primary
afmRecoveryUseFset Is used with the mmchfileset and mmcrfileset commands. Enables the specified AFM fileset to use the custom storage path, which the afmRecoveryDir parameter specifies at the cluster level, for the recovery or reconcile instead of the default location. During the recovery, if resynchronization or reconcile operations are triggered on the AFM fileset, AFM uses the /var/mmfs/afm/ default location to store the recovery data. AFM uses these files to synchronize the pending data to the target. For the AFM fileset to use the custom storage path, which the afmRecoveryDir parameter specifies, AFM uses different paths based on a set recovery parameter or parameters.	no	yes no	Boolean	Fileset		RO, IW, SW, LU	RO, IW, SW, LU	Primary
afmRefreshAsync Cache data refresh operation in asynchronous mode improves performance of applications by querying data. Specify the value as 'yes' for the cache data refresh operation to be in asynchronous mode. In the IW mode, a revalidation request for files or directories is queued as an asynchronous request, and data in the last synchronized state is returned. The data is refreshed at cache after revalidation with home is complete. Revalidation time depends on the network availability and bandwidth.	no	yes no	Boolean	Cluster, fileset		IW, LU, RO, SW		
afmRefreshOnce Enables AFM to perform revalidation on files and directories only one time. This parameter improves the application performance after the data migration to the cache and the application is started on the cache data. When this parameter is set to yes, files and directories revalidation is performed only one time. Therefore, only one revalidation request goes to the home or target.	no	yes no	Boolean	Fileset		IW, LU, RO		

Table 10. AFM, AFM-DR, and AFM to cloud object storage configuration parameters (continued)

Parameter ¹	Default value ²	Valid value ³	Unit ⁴	Application ⁵	Parallel transfer ⁶	AFM file cache ⁷	AFM object cache ⁸	AFM-DR ⁹
afmRecoveryVer2 Enables optimized version of the recovery on the AFM Single Writer or AFM DR Primary filesset. After the optimized recovery version is set, AFM avoids using the older version of the recovery and uses the version V2 to identify the remove and rename operations. The new recovery version identify the home directory entries by running the GPFS policy at the home when recovery event is triggered.	disabled	yes no	Boolean	Fileset		SW		Primary
afmResyncVer2 Increases the replication performance and scalability of a system that has heavy stress and workloads. It increases the message queuing performance by using an on-demand dependency resolution for queued messages, in case of recovery and/or resync is running.	no	yes no	Boolean	Fileset	Yes	SW		Primary
afmRevalOpWaitTimeout Specifies the time that AFM waits for completion of revalidation to get response from the home cluster. Revalidation checks if any changes are available at home (data and metadata) that need to be updated to the cache cluster. Revalidation is performed when application trigger operations like lookup or open at cache. If revalidation is not completed within this time, AFM cancels the operation and returns data available at cache to the application.	180	5 - 2147483647	Second	Cluster		IW, RO		
afmRPO Specifies the recovery point objective (RPO) time interval for an AFM DR filesset in minutes. Disabled is the default value of the parameter. You can also specify the value with the suffix M for minutes, H for hours, or W for weeks. For example, for 12 hours specify 12H. If you do not add a suffix, the value is assumed to be in minutes.	disabled	60 - 2147483647	Minutes, hours, weeks	Cluster, filesset				Primary
afmSecondaryRW Specifies if the secondary is read-write or not.	no	yes no	Boolean	Cluster				Secondary
afmShowHomeSnapshot Controls the visibility of the home snapshot directory in the cache cluster. For this to be visible in the cache cluster, this variable must be set to yes, and the snapshot directory name in the cache and home clusters must not be the same.	no	yes no	Boolean	Cluster, filesset		LU, RO		
afmSyncNFSv4ACL Enables the migration of NFSv4 ACLs from third-party file systems. When this parameter is enabled, data from a third-party storage is migrated to IBM Storage Scale.	0	0 1	Number	Cluster	Yes	IW, LU, RO, SW		Primary, secondary

Table 10. AFM, AFM-DR, and AFM to cloud object storage configuration parameters (continued)

Parameter ¹	Default value ²	Valid value ³	Unit ⁴	Application ⁵	Parallel transfer ⁶	AFM file cache ⁷	AFM object cache ⁸	AFM-DR ⁹
afmSyncOpWaitTimeout Specifies the time that AFM waits for completion of any inflight synchronous operation that is synchronizing with the home or the primary cluster. When any application is performing any synchronous operation on the home or primary cluster, AFM tries to get a response from the home or the primary cluster. If the home or the primary cluster is not responding, application might be unresponsive. If operation does not complete in this timeout interval, AFM cancels the operation.	180	5 - 2147483647	Second	Cluster		IW, RO, SW		Primary
afmSyncReadMount Creates a separate mount path at the gateway node through which synchronous read calls are sent to the home cluster separately. This arrangement helps other operations that are running on the AFM fileset to synchronize with the home cluster by using the default mount path. If the home cluster is responding slow or busy, the synchronous read operations request that are coming through separate mount path at the gateway node gets killed and the read request is re-tried without affecting the other operations as the current operation is performed on the separate mount path. In case of AFM (SW/IW) fileset , when you enable the afmSyncReadMount parameter, it avoids dropping the whole read request queue during a busy or slow home response, thus avoid sending EIO back to the application which is performing the read operation at the Cache cluster. This behavior skip moving AFM fileset into a stale state and avoid recovery process for AFM SW/IW fileset. You can enable afmSyncReadMount parameter on the AFM Single-Writer (SW), Independent-Writer(IW), Local-Update(LU), and Read-Only(RO) mode fileset by using mmcrfileset or mmchfileset command. The valid values for the afmSyncReadMount parameter are "yes" or "no". To set or unset the afmSyncReadMount parameter on an AFM Single-Writer or AFM Independent-Writer mode fileset, you must stop or unlink the fileset. For more information, see <i>Stop and start replication on a fileset in IBM Storage Scale: Concepts, Planning, and Installation Guide</i> .	no	yes no	Boolean	fileset	IW, SW, RO, LU			

Table 10. AFM, AFM-DR, and AFM to cloud object storage configuration parameters (continued)

Parameter ¹	Default value ²	Valid value ³	Unit ⁴	Application ⁵	Parallel transfer ⁶	AFM file cache ⁷	AFM object cache ⁸	AFM-DR ⁹
afmTarget Identifies the home that is associated with the cache. The only allowed value is disable. It is used to convert AFM filesets to regular independent fileset. <code>mmchfileset fs1_ro -p afmTarget=disable</code> After an AFM fileset is converted to a regular fileset, the fileset cannot be changed back to an AFM fileset.	disable		String	Fileset				

¹ Lists the name of an AFM configuration parameter and its description.² Lists a default value for a parameter.³ Lists acceptable values for a parameter.⁴ Lists a unit of measurement for a parameter.⁵ Indicates if a parameter can be applied across a cluster, to a specific fileset or both cluster and fileset. A value of both implies there can be a cluster-wide value but individual filesets may have a different value.⁶ Indicates if a parameter impacts parallel data transfers.⁷ Lists the file cache types to which a parameter can apply.⁸ Indicates if a parameter applies to an object cache (DAAA).⁹ Indicates if a parameter impacts a primary cluster, a secondary cluster or both clusters.

Configuration changes in an existing AFM relationship

You can modify an existing AFM relation by making a few changes in the configuration.

Adding gateway nodes to the cache cluster

You can add a gateway node or remove the existing gateway nodes by using the **mmchnode --gateway| --nogateway** command.

After gateway node addition or removal, AFM might reassign the existing filesets to gateway nodes based on the **afmHashVersion**.

Ensure that AFM fileset queues are empty, or existing gateway nodes are shut down before you run the **mmchnode** command.

If the gateway node is added or removed with all the gateway nodes in the Active state, the cluster might readjust and might appear unresponsive for some time, depending on the configuration. If the cluster remains unresponsive, restart the gateway node.

You can remove or add a gateway node in the parallel data transfer mapping.

- Deletion of a gateway node fails, if it is a part of the mapping. You must run the **mmafmconfig update** command to remove the IP address or the node name from the mapping list. You can then run the **mmchnode --nogateway** command to remove the gateway node role.
- To add a gateway role to a node, you must first run the **mmchnode --gateway** command followed by the **mmafmconfig update** command to add the IP address or the node name to the mapping list.

The NFS server at the home cluster

The NFS server, the mount path, or the IP address at the home cluster can be changed.

The existing AFM filesets on cache must be updated to point to the new target. The home cluster and file system do not change. Therefore, any change can be reflected in the cache cluster by using the **mmafmctl failover** command with the **--target-only** option.

However, if you change an NFS server, the new NFS server must be in the same home cluster and must have the same architecture as the existing NFS server in the target path. In other cases, the failover must be performed without the `--target-only` option. If the target protocol changes from NSD to NFS or vice-versa, the `mmadmctl failover` command must be used without the `--target-only` option.

For more information, see the *Changing home of AFM cache* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Enabling AFM Network File System version 4

Enable AFM NFSv4 support by using the `mmchconfig` command.

1. Enable the `afmNFSVersion` parameter.

```
# mmchconfig afmNFSVersion=4.1 -i
```

A sample output is as follows:

```
mmchconfig: Command successfully completed
mmchconfig: Propagating the cluster configuration data to all affected nodes. This is an
asynchronous process.
0YO! c25m4n01[01:13:21][~]:Thu Jul 15 01:13:21 EDT 2021: mmcommon pushSdr_async: mmsdrfs
propagation started
Thu Jul 15 01:13:23 EDT 2021: mmcommon pushSdr_async: mmsdrfs propagation completed; mmdsh
rc=0
```

2. Check the NFS version.

```
# mmlsconfig afmNFSVersion
```

A sample output is as follows:

```
afmNFSVersion 4.1
```

3. Enable the `afmSyncNFSV4ACL` parameter.

```
# mmchconfig afmSyncNFSV4ACL=1 -i
```

A sample output is as follows:

```
mmchconfig: Command successfully completed
mmchconfig: Propagating the cluster configuration data to all affected nodes. This is an
asynchronous process.
0YO! c25m4n01[01:13:57][~]:Thu Jul 15 01:13:57 EDT 2021: mmcommon pushSdr_async: mmsdrfs
propagation started
Thu Jul 15 01:13:59 EDT 2021: mmcommon pushSdr_async: mmsdrfs propagation completed; mmdsh
rc=0
```

4. Check the `afmSyncNFSV4ACL` parameter value.

```
# mmlsconfig afmSyncNFSV4ACL
```

A sample output is as follows:

```
afmSyncNFSV4ACL 1
```

For more information about these parameters, see the `mmchconfig` command in *IBM Storage Scale: Command and Programming Reference Guide*.

Mapping IDs with the AFM Network File System version 4

If Active File Management (AFM) is configured with Network File System version 4 (NFSv4) as a replication protocol, all applications or protocol nodes in the cache cluster must have access to the Lightweight Directory Access Protocol (LDAP) server. For example, for correct ID mapping, access to an Active Directory (AD). Otherwise, you can disable the ID mapping.

Ensure that both client and server have matching UIDs and GIDs even with NFSv4. The ID mapping is done to map an ID to a name and vice-versa. If the ID mapping is disabled, NFS clients send numeric UIDs or GIDs in outgoing attribute calls, and NFS servers send numeric UIDs or GIDs in outgoing attribute replies. If NFS clients send numeric UIDs or GIDs in a SETATTR call, they receive an NFS4ERR_BADOWNER reply from the NFS server. Clients re-enable the ID mapping and send user@domain strings for that a specific mount henceforth.

1. Disable ID mapping.

a) Disable ID mapping with the Knfs protocol.

The ID mapping does not manage:

- On an NFS client

```
# echo 'Y' > /sys/module/nfs/parameters/nfs4_disable_idmapping
```

- On an NFS server

```
# echo 'Y' > /sys/module/nfssd/parameters/nfs4_disable_idmapping
```

b) Disable ID mapping with the Ganesha protocol.

i) Copy the configuration file.

```
# cp /var/mnfs/ces/nfs-config/gpfs.ganesha.main.conf /tmp
```

ii) Open the /tmp/gpfs.ganesha.main.conf file and add the following information, and then save it.

```
NFSv4
{
    delegations=FALSE;
    domainname=virtual1.com;
    Only_Numeric_Owners=TRUE;                                <-- Add Only_Numeric_Owners option
    grace_period=90;
    lease_lifetime=60;
    minor_versions=0,1;
}
```

iii) Update the configuration file permanently.

```
# mmccr fput gpfs.ganesha.main.conf /tmp/gpfs.ganesha.main.conf
```

iv) Stop and start the cluster export services.

```
# mmces service stop nfs -a
```

```
# mmces service start nfs -a
```

2. Add a domain name to an NFS client and an NFS server.

- On an NFS server, modify the /etc/idmapd.conf file with a proper domain (FQDN).

a. Change the NFS server configuration.

```
# mnfs config change "IDMAPD_DOMAIN=storage1test.domain.com"
```

A sample output is as follows:

```
mnfs: The NFS configuration was changed successfully.
mnfs: NFS server restarted on all NFS nodes on which NFS server is running.
```

b. Verify the configuration.

```
# mnfs config list
```

A sample output is as follows:

```
NFS Ganesha Configuration
=====
DELEGATIONS: FALSE
DOMAINNAME: VIRTUAL1.COM
GRACE_PERIOD: 90
LEASE_LIFETIME: 60
...
Imapd Configuration
=====
DOMAIN: STORAGE1TEST.TUC.STGLABS.IBM.COM
LOCAL_REALMS: localdomain
=====
```

- On an NFS client, set a domain in the /etc/idmapd.conf file.

- a. Issue the **mmdsh** command on the multiple nodes.

```
# mmdsh -N prt001st003,prt002st003,prt003st003
```

- b. Check the contents of the file.

```
# cat /etc/idmapd.conf | grep storage
```

A sample output is as follows:

```
prt001st003: Domain = storage1test.tuc.stglabs.ibm.com
prt003st003: Domain = storage1test.tuc.stglabs.ibm.com
prt002st003: Domain = storage1test.tuc.stglabs.ibm.com
```

- c. Restart the idmapd service.

```
# systemctl restart nfs-idmapd.service
```

Chapter 13. Configuring AFM-based DR

The following topics list the parameters that are necessary to configure AFM-based DR.

Changing configuration in an existing AFM DR relationship

See the following examples of changing gateway nodes and an NFS server:

Changing NFS server on secondary

The NFS server, or mount path or IP address on secondary can change.

Existing AFM primary filesets need to be updated to point to the new target. As the secondary cluster and file system do not change, any of these changes can be reflected in the cache by using the **mmafmctl** command with the **changeSecondary -target-only** option. If the NFS server changes, the new NFS server must be in the same secondary cluster and the architecture must be the same as the existing NFS server in the target path. If the NFS server is not in the same secondary cluster or the architecture is not the same, the **changeSecondary** must be performed with the **--inband** option. If the target protocol changes from NSD to NFS or vice-versa, the **mmafmctl changeSecondary** command must be used with **--inband** option.

Changing gateway nodes on primary

You can add new gateway or remove existing gateway nodes by using the **mmchnode** command. AFM automatically adjusts the existing filesets to use the latest configured gateways.

You must shut down all the existing gateway nodes and then add or remove gateway by using **mmchnode** command on a cluster that is running applications on AFM DR filesets. If the gateway nodes are changed while all gateway nodes are active, the gateway nodes might not be responding, or cluster-wide waiters might be observed after you run the **mmchnode** command. Recycle the active gateway nodes.

It is not possible to remove existing gateway nodes if they are part of a mapping. You can remove the gateway nodes from the mapping by using **mmafmconfig** command.

Note: Whenever you add or remove a gateway node, ensure that you update the list of IP addresses in the export map on the home.

Chapter 14. Configuring AFM to cloud object storage

IBM Storage Scale provides capability to use the AFM to cloud object storage for files and objects. An IBM Storage Scale cluster can be configured to create AFM to cloud object storage filesets to connect to a cloud object storage such as Amazon S3, IBM Cloud Object Storage, Microsoft Azure Blob, or services that use S3 APIs.

Ensure that the following conditions are met before you configure the AFM to cloud object storage:

- IBM Storage Scale cluster is up and running with IBM Storage Scale 5.1.0 or later. For more information, see *Steps for establishing and starting your IBM Storage Scale cluster and GPFS cluster creation considerations* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
- Gateway nodes are provided in the IBM Storage Scale cluster to manage the AFM to cloud object storage replication.
- A cloud object storage is provided with endpoints with required storage and required access and secret keys.
- HTTP or HTTPS protocols are configured on a cloud object storage.
- Security and firewall settings are configured properly for seamless connectivity between an IBM Storage Scale cluster and a cloud object storage.

Considerations

- An IBM Storage Scale cluster hosts an AFM to cloud object storage fileset and a cloud object storage that are two different entities. The fileset and the cloud object storage are connected over internet or WAN.
- There are no special considerations on the cluster setup for using the AFM to cloud object storage functions. AFM to cloud object storage functions are available in all editions of IBM Storage Scale. You can install IBM Storage Scale and deploy protocols either manually or by using the installation toolkit. For more information, see *Steps for establishing and starting your IBM Storage Scale cluster* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
- After all nodes are upgraded to the new code, you must finalize the upgrade. For more information about the upgrade, see *Completing the upgrade to a new level of IBM Storage Scale* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
- Nodes must be identified on the AFM to cloud object storage cluster that can act as gateway nodes. Gateway nodes can be configured in the cache cluster before you create filesets and start applications.
- In case of AFM to cloud object storage, backend services are accessible but most endpoints are not able to ping from the AFM cluster. A user with an administrator privilege can set **afmSkipHomePing** parameter to yes (`mmchconfig afmSkipHomePing=yes`). After the configuration (`mmchconfig afmSkipHomePing=yes`) is completed, this AFM to cloud object storage will not ping the endpoints and the fileset will not go into disconnected state.

For the AFM to cloud object storage configuration, you can use the **mmafmcosaccess**, **mmafmcosctl**, **mmafmcosconfig**, and **mmafmcoskeys** commands. After the access and secret keys are provided on the cloud object storage endpoints, these commands can be directly used to configure the AFM to cloud object storage. For more information about the cloud object storage commands, see *mmafmcosaccess*, *mmafmcosconfig*, *mmafmcosctl*, and *mmafmcoskeys* commands in the *IBM Storage Scale: Command and Programming Reference Guide*.

1. Ensure that at all nodes in a cluster are up and running with IBM Storage Scale 5.1.0 and the file system level is 5.1.0 or later.

- a) To get the cluster information, issue the following command:

```
# mmclscluster
```

A sample output is as follows:

```

GPFS cluster information
=====
GPFS cluster name: afm2cos.cluster
GPFS cluster id: 6470607479877415271
GPFS UID domain: afm2cos.cluster
Remote shell command: /usr/bin/ssh
Remote file copy command: /usr/bin/scp
Repository type: CCR

GPFS cluster configuration servers:
-----
Primary server: Node3
Secondary server: Node4

Node Daemon node IP address Admin node Designation
name name
-----
1 Node2 192.168.105.62 Node2 quorum-manager
2 Node5 192.168.105.65 Node5 quorum-gateway
3 Node4 192.168.105.64 Node4 quorum-gateway
4 Node3 192.168.105.63 Node3 quorum-manager

```

Note: In this cluster, the Node4 and Node5 gateway nodes are provided.

- To check the nodes state, issue the following command:

```
# mmgetstate -a
```

A sample output is as follows:

```

Node number Node name GPFS state
-----
1           Node2   active
2           Node5   active
3           Node4   active
4           Node3   active

```

- To get the file system information, issue the following command:

```
# mmfs fs1 -V -T
```

A sample output is as follows:

```

flag  value          description
-----
-V    24.00 (5.1.0.0) File system version
-T    /gpfs/fs1       Default mount point

```

- Configure cloud object storage endpoints. These endpoints can be configured by using the management or configuration system that each cloud object storage provides.

```

Cloud Object Server Endpoint : http://192.168.118.121
ACCESS_KEY : myexampleaccesskey1234$#
SECRET_KEY : myexamplesecretkey1234567890#*
Endpoint Port Number : 80

```

A bucket can be created on a cloud object storage before you configure a fileset or it can be created from the IBM Storage Scale cluster.

- Configure the AFM to cloud object storage by using the keys that are created or generated on the cloud object storage.

- To store the keys on the buckets in the IBM Storage Scale cluster, issue the following command:

```
# mmafmcoskeys afmtocos1 set myexampleaccesskey1234$ myexamplesecretkey1234567890#*
```

Note: The bucket can exist on the cloud object storage or can be created from the AFM to cloud object storage setup.

Where:

afmtocos1

Name of the bucket

Access key

myexampleaccesskey1234\$

Secret key

myexamplesecretkey1234567890#

- b) To check that the correct keys are set on the bucket, you can issue the following command:

```
# mmamfmcoskeys afmtocos1 get myexampleaccesskey1234$:myexamplesecretkey1234567890#
```

4. After the keys are set, to configure the AFM to cloud object storage, issue the following command:

```
# mmamfmcosconfig fs1 afmtocos1 --endpoint http://192.168.118.121 --uid 0 --gid 0 --new-bucket afmtocos1 --mode iw --object-fs
```

Where:

fs1

Name of a file system.

afmtocos1

Name of a fileset name.

iw

The independent writer mode of the AFM to cloud object storage fileset.

Note: This command is run on the Node5 node, which becomes a gateway node for the afm2cos1 fileset.

After the AFM to cloud object storage setup is done, you can see information about a relation fileset by issuing the following command:

```
# mmclsfileset fs1 afmtocos1 --afm -L
```

A sample output is as follows:

```
Filesets in file system 'fs1':  
Attributes for fileset afmtocos1:  
=====
```

Status	Linked
Path	/gpfs/fs1/afmtocos1
Id 1	
Root inode	524291
Parent Id	0
Created	Wed Sep 9 05:21:15 2020
Comment	
Inode space	1
Maximum number of inodes	100352
Allocated inodes	100352
Permission change flag	chmodAndSetacl
afm-associated	Yes
Target	http://192.168.118.121:80/afmtocos1
Mode	independent-writer
File Lookup Refresh Interval	120
File Open Refresh Interval	120
Dir Lookup Refresh Interval	120
Dir Open Refresh Interval	120
Async Delay	15 (default)
Last pSnapId	0
Display Home Snapshots	no
Parallel Read Chunk Size	0
Number of Gateway Flush Threads	4
Prefetch Threshold	0 (default)
Eviction Enabled	yes (default)
Parallel Write Chunk Size	0
IO Flags	0x4000000 (afmObjectSubdir)

5. Create some objects by using the AFM to cloud object storage fileset.

```
# dd if=/dev/urandom of=/gpfs/fs1/afmtocos1/object1 count=4 bs=256K
# dd if=/dev/urandom of=/gpfs/fs1/afmtocos1/object2 count=8 bs=256K
# dd if=/dev/urandom of=/gpfs/fs1/afmtocos1/object3 count=12 bs=256K
```

These created objects are replicated to the cloud object storage asynchronously and the cache state is dirty, then they are being replicated.

To check the cache state, issue the following command:

```
# mmafmctl fs1 getstate
```

A sample output is as follows:

Fileset	Name	Fileset	Target	Cache State	Gateway	Node	Queue	Length	Queue	numExec
afmtocos1			http://192.168.118.121:80/afmtocos1	Dirty	c7f2n05		3		6	

When all the operations or objects that are created are synced to a cloud object storage, the cache state becomes Active. To check the cache state, issue the following command:

```
# mmafmctl fs1 getstate
```

A sample output is as follows:

Fileset	Name	Fileset	Target	Cache State	Gateway	Node	Queue	Length	Queue	numExec
afmtocos1			http://192.168.118.121:80/afmtocos1	Dirty	c7f2n05		0		9	

To get the fileset contents, issue the following command:

```
# ls -lsh /gpfs/fs1/afmtocos1
```

A sample output is as follows:

```
total 5.0M
1.0M -rw-r--r-- 1 root root 1.0M Sep 9 05:27 object1
2.0M -rw-r--r-- 1 root root 2.0M Sep 9 05:28 object2
3.0M -rw-r--r-- 1 root root 3.0M Sep 9 05:28 object3
```

- Check that objects are replicated and synchronized with a cloud object storage by using different APIs or GUI that the cloud object storage provides.

Example:

```
<xml>
Name : afmtocos1/
Date : 2020-09-09 05:20:17 EDT
Size : 0 B
Type : Bucket
Name : object1
Date : 2020-09-09 05:24:17 EDT
Size : 1.0 Mib
ETag : 36b30c1b8016f0cc4ca41bec0d12f588
Type : file
Metadata :
Content-Type: application/octet-stream
Name : object2
Date : 2020-09-09 05:24:34 EDT
Size : 2.0 Mib
ETag : 9d17d1fd443287a83445c4616864eb72
Type : file
Metadata :
Content-Type: application/octet-stream
Name : object3
Date : 2020-09-09 05:24:44 EDT
Size : 2.0 Mib
ETag : e8b894cf47871a56f0a9c48bc99bfea6
Type : file
Metadata :
Content-Type: application/octet-stream
```

- Read the object that is created on the cloud object storage on the AFM to cloud object storage fileset.

In the following example, objectcreatedonCOS1, objectcreatedonCOS2, and objectcreatedonCOS3 are new objects that are created on the cloud object storage:

```
1.0MiB object1
2.0MiB object2
2.0MiB object3
1.0MiB objectcreatedonCOS1
2.0MiB objectcreatedonCOS2
3.0MiB objectcreatedonCOS3
```

To get contents of a fileset on the IBM Storage Scale cluster, issue the following command:

```
# ls -lsh /gpfs/fs1/afmtocos1
```

A sample output is as follows:

```
total 5.0M
1.0M -rw-r--r-- 1 root root 1.0M Sep 9 05:27 object1
2.0M -rw-r--r-- 1 root root 2.0M Sep 9 05:28 object2
2.0M -rw-r--r-- 1 root root 2.0M Sep 9 05:28 object3
0 -rwx----- 1 root root 1.0M Sep 9 2020 objectcreatedonCOS1
0 -rwx----- 1 root root 2.0M Sep 9 2020 objectcreatedonCOS2
0 -rwx----- 1 root root 3.0M Sep 9 2020 objectcreatedonCOS3
```

Now you can see the object metadata in the AFM to cloud object storage fileset and its contents are not cached. When the objects are read, all the data is pulled in from the cloud object storage. This is on demand from applications that are hosted on IBM Storage Scale.

An example of reading the objects by the applications is as follows:

```
# cat /gpfs/fs1/afmtocos1/objectcreatedonCOS1 > /dev/null
# cat /gpfs/fs1/afmtocos1/objectcreatedonCOS2 > /dev/null
# cat /gpfs/fs1/afmtocos1/objectcreatedonCOS3 > /dev/null
```

To get the contents a fileset, issue the following command:

```
# ls -lsh /gpfs/fs1/afmtocos1
```

A sample output is as follows:

```
total 11M
1.0M -rw-r--r-- 1 root root 1.0M Sep 9 05:27 object1
2.0M -rw-r--r-- 1 root root 2.0M Sep 9 05:28 object2
2.0M -rw-r--r-- 1 root root 2.0M Sep 9 05:28 object3
1.0M -rwx----- 1 root root 1.0M Sep 9 2020 objectcreatedonCOS1
2.0M -rwx----- 1 root root 2.0M Sep 9 2020 objectcreatedonCOS2
3.0M -rwx----- 1 root root 3.0M Sep 9 2020 objectcreatedonCOS3
```

8. Download the objects that are created on a cloud object storage on priority or preference by using the **mmafmcosctl download** command.

- a) Create new objects with a .imp extension on a cloud object storage.

```
1.0MiB object1
2.0MiB object2
2.0MiB object3
1.0MiB objectcreatedonCOS1
1.0MiB objectcreatedonCOS1.imp
2.0MiB objectcreatedonCOS2
2.0MiB objectcreatedonCOS2.imp
3.0MiB objectcreatedonCOS3
3.0MiB objectcreatedonCOS3.imp
```

- b) Download or prefetch the object list that is created on the IBM Storage Scale cluster by issuing the following command:

```
# cat ObjectList
```

A sample output is as follows:

```
/gpfs/fs1/afmtocos1/objectcreatedonCOS1.imp  
/gpfs/fs1/afmtocos1/objectcreatedonCOS2.imp  
/gpfs/fs1/afmtocos1/objectcreatedonCOS3.imp
```

c) To download the objects, issue the **mmafmcosctl** command:

```
# mmafmcosctl fs1 afmtocos1 /gpfs/fs1/afmtocos1/ download --object-list ObjectList --data
```

A sample output is as follows:

```
Queued (Total) Failed TotalData  
                           (approx in Bytes)  
0      (0)      0      0  
3      (0)      0    6291456  
Object Downloads successfully queued at the gateway.
```

d) To check the cache state, issue the following command:

```
# mmafmctl fs1 getstate
```

A sample output is as follows:

Fileset Name	Fileset Target	Cache State	Gateway Node	Queue Length	Queue numExec
afmtocos1	http://192.168.118.121:80/afmtocos1	Active	c7f2n05	0	71

e) To get the contents of the fileset, issue the following command:

```
# ls -lsh /gpfs/fs1/afmtocos1
```

A sample output is as follows:

```
total 17M  
1.0M -rw-r--r-- 1 root root 1.0M Sep 9 05:27 object1  
2.0M -rw-r--r-- 1 root root 2.0M Sep 9 05:28 object2  
2.0M -rw-r--r-- 1 root root 2.0M Sep 9 05:28 object3  
1.0M -rwx----- 1 root root 1.0M Sep 9 2020 objectcreatedonCOS1  
1.0M -rwx----- 1 root root 1.0M Sep 9 2020 objectcreatedonCOS1.imp  
2.0M -rwx----- 1 root root 2.0M Sep 9 2020 objectcreatedonCOS2  
2.0M -rwx----- 1 root root 2.0M Sep 9 2020 objectcreatedonCOS2.imp  
3.0M -rwx----- 1 root root 3.0M Sep 9 2020 objectcreatedonCOS3  
3.0M -rwx----- 1 root root 3.0M Sep 9 2020 objectcreatedonCOS3.imp
```

Note: With the **objectFS** mode, objects can be read on demand from a cloud object storage. Whereas the **ObjectOnly** mode download and upload can be used for priority data sync depending upon the mode of the AFM to cloud object storage fileset.

Configuring an AFM to cloud object storage fileset with Microsoft Azure Blob

Complete this procedure to set up an AFM to cloud object storage fileset with Microsoft Azure Blob.

Ensure that the following conditions are met:

1. Microsoft Azure Blob service is provisioned with storage accounts and respective containers under it.
2. When setting up storage accounts a preferred region is selected.
3. Access keys and endpoint information is available for associating it with an AFM to cloud object storage fileset.
4. An IBM Storage Scale cluster is set up and provisioned with gateway nodes that have connectivity to Microsoft Azure Blob endpoints.
5. `gpfs.afm.cos*.rpm` is present on gateway nodes.

Note: The **mmafmcoskeys** command needs bucket name, access key, and secret key parameters. For the Microsoft Azure Blob support, a bucket name is the container name, an access key (akey) is name of a storage account, and a secret key is the key obtained from the storage accounts security section.

1. Set up Microsoft Azure Blob services, storage account, and note the endpoint and respective keys.
2. Add the endpoint and keys into an IBM Storage Scale cluster by issuing the following command:

```
# mmafmcoskeys fileset1 set afmtest Ga3WljrCeUTROPORi2IAaUMb*****K5husMUb01iQtd8xwDni/
UMf64bhwtYHW5vv2sb0YgEK8R6+ASTaYjRLw==
```

where,

fileset1

Is the name of a container that will be created on Microsoft Azure Blob.

afmtest

Is the name of a storage account under which the *fileset1* container will be created.

3. Set up AFM to cloud object storage relationship.

```
# mmafmcosconfig fs1 fileset1 --endpoint https://afmtest.blob.core.windows.net --xattr --new-
bucket fileset1 --mode iw --object-fs --azure --acl --directory-object
```

where,

--azure

Creates the AFM to cloud object storage relationship with Microsoft Azure Blob as a backend.

fs1

A file system name where the *fileset1* fileset will be created.

--directory-object

Supports the directory structure during the creation of the AFM to cloud object relationship.

Note: Microsoft Azure Blob creates the no_name file in directories for internal usage. Do not delete this file from the cloud object storage.

4. Verify the fileset parameters.

```
# mmclsfileset fs1 fileset1 --afm -L
```

A sample output is as follows:

```
Filesets in file system 'fs1':  
  
Attributes for fileset fileset1:  
=====  
Status  
Path  
Id  
Root inode  
Parent Id  
Created  
Comment  
Inode space  
Maximum number of inodes  
Allocated inodes  
Permission change flag  
afm-associated  
Permission inherit flag  
Target  
Mode  
File Lookup Refresh Interval  
File Open Refresh Interval  
Dir Lookup Refresh Interval  
Dir Open Refresh Interval  
Async Delay  
Last pSnapId  
Display Home Snapshots  
Parallel Read Chunk Size  
Number of Gateway Flush Threads  
Prefetch Threshold  
Eviction Enabled  
Parallel Write Chunk Size  
Linked  
/gpfs/fs1/fileset1  
207  
5767171  
0  
Fri Oct 20 01:52:27 2023  
11  
100352  
100352  
chmodAndSetacl  
Yes  
inheritAclOnly  
https://afmtest.blob.core.windows.net:443/fileset1  
independent-writer  
120  
120  
120  
120  
1  
(default)  
0  
no  
0  
4  
0  
(default)  
yes  
(default)  
0
```

```

IO Flags          0x8280000
  (afmObjectXattr,afmObjectDirectoryObj,afmObjectACL)
IO Flags2        0x8280800 (afmObjectAZ)

```

5. Create some data.

```

Node1 0] dd if=/dev/urandom of=/gpfs/fs1/fileset1/object1 bs=256K count=4
4+0 records in
4+0 records out
1048576 bytes (1.0 MB, 1.0 MiB) copied, 0.00787632 s, 133 MB/s

Node1 0] dd if=/dev/urandom of=/gpfs/fs1/fileset1/object2 bs=256K count=4
4+0 records in
4+0 records out
1048576 bytes (1.0 MB, 1.0 MiB) copied, 0.00597403 s, 176 MB/s

Node1 0] dd if=/dev/urandom of=/gpfs/fs1/fileset1/object3 bs=256K count=4
4+0 records in
4+0 records out
1048576 bytes (1.0 MB, 1.0 MiB) copied, 0.00683723 s, 153 MB/s

```

6. Verify that the created data is placed on the Microsoft Azure Blob storage.

```
# ls -lash /gpfs/fs1/fileset1/
```

A sample output is as follows:

```

total 3.3M
 512 drwxrws---    5 root  root 4.0K Oct 20 01:56 .
256K drwxr-xr-x   13 root  root 256K Oct 20 01:52 ..
 512 drwx----- 65535 root  root 4.0K Oct 20 01:52 .afm
1.0M -rw-r--r--    1 root  root 1.0M Oct 20 01:56 object1
1.0M -rw-r--r--    1 root  root 1.0M Oct 20 01:56 object2
1.0M -rw-r--r--    1 root  root 1.0M Oct 20 01:56 object3
 512 drwx----- 65535 root  root 4.0K Oct 20 01:52 .pconflicts
 512 drwx----- 65535 root  root 4.0K Oct 20 01:52 .ptrash
 512 dr-xr-xr-x    2 root  root 8.0K Dec 31 1969 .snapshots

```

7. Check the data queue on the gateway node by issuing the following command:

```
# mmamfctl fs1 getstate
```

A sample output is as follows:

Fileset Name	Fileset Target	Cache State	Gateway
Node Queue	Length Queue numExec		
fileset1 c7f2n09	https://afmtest.blob.core.windows.net:443/fileset1 0 10	Active	

When all the data is pushed from the queue to Microsoft Azure Blob storage, it will be visible in containers. The following figure shows files present in the *fileset1* container.

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
object1	10/20/2023, 11:26:16...	Hot (Inferred)		Block blob	1 MiB	Available
object2	10/20/2023, 11:26:29...	Hot (Inferred)		Block blob	1 MiB	Available
object3	10/20/2023, 11:26:36...	Hot (Inferred)		Block blob	1 MiB	Available

Configuring AFM to cloud object storage for Azure Blob storage by using MinIO as S3 gateway

The IBM Storage Scale cluster can be configured to create AFM to cloud object storage filesets to connect to a cloud object storage such as Azure Blob by using MinIO as S3 gateway for communication.

1. Set up Azure Blob storage and configure the container and buckets.
2. Obtain an Access key.
3. Start S3 gateway by issuing the following command:

```
# export MINIO_ROOT_USER=<container_name>
# export MINIO_ROOT_PASSWORD=<Access_Key>
# ./minio gateway azure
```



Attention: The S3 gateway functionality of MinIO must be run on the older supported version only. The latest version of MinIO has deprecated the support of using MinIO as S3 gateway for Azure Blob Storage. For more information, see [Gateway Deprecation Implications for Azure Customers](#).

4. Setup keys with IBM Storage Scale AFM to cloud object storage.

```
# mmafmcoskeys bkt1:<S3_Gateway_IP> set <container> <Access_Key>
```

5. Create AFM to cloud object storage that points to the bucket (in step 4) with set credentials.

```
# mmafmcosconfig <FS> <fileset_name> --endpoint http://<S3_GW_IP>:<port> --object-fs --xattr --new-bucket bkt1 --mode sw --acls
```

6. Start the application in AFM to cloud object storage.

Configuring AFM to cloud object storage fileset by using use-keys and STS token

You can configure AFM to cloud object storage fileset by using use-keys option and Security Token Service (STS).

An AFM to cloud object storage fileset can be configured either by providing the credentials that includes use-keys and security token service (STS) to AFM or by configuring a file to return the credentials.

The **mmafmcoskeys** command can be used to store the credentials with AFM. AFM stores and uses the credentials for communication with the server.

1. To configure AFM to retrieve the credentials without storing it, you must configure the file /var/mmfs/etc/mmuid2keys to return the credentials after execution. This mmuid2keys file must be available at all the AFM gateway nodes and must have root executable permission enabled.
2. When you start to retrieve the credentials from mmuid2keys file, --user-keys option must be specified while creating fileset by using the **mmafmcosconfig** command.

The following example shows how to configure AFM to cloud object storage fileset by using --use-keys option and Security Token Service (STS):

1. Obtain temporary credentials.
2. Update the /var/mmfs/etc/mmuid2keys file to return the credentials in following format as shown after the execution on all the gateway nodes.

```
#cat /var/mmfs/etc/mmuid2keys
echo "akey:skey:sts"
#chmod +x /var/mmfs/etc/mmuid2keys
```

3. Create a fileset by using **--user-keys** parameter with the **mmafmcosconfig** command.

```
#mmafmcosconfig fs1 sw1 --endpoint http://s3.amazonaws.com --user-keys --bucket bkt1 --mode sw1 --xattr --acl --debug --cleanup
```

Configuring AFM to cloud object storage to use Google cloud storage

You need to configure AFM to cloud object storage to use features of Google cloud storage.

The AFM to cloud object storage filesets can be configured to communicate to the Google Cloud Object Storage by specifying **-gcs** option when you create the fileset. For more information about AFM to cloud object storage configuration, see [Chapter 14, “Configuring AFM to cloud object storage,” on page 151](#).

Do the following steps to create a fileset:

1. Create a bucket at the Google Cloud Object Storage server.
For example, 'bkt1' is the name of the bucket.
2. Setup access key and secret key with AFM.

```
mmafmcoskeys bkt1:us-west2@storage.googleapis.com set <access_key> <secret_key>
```

3. Create AFM to cloud object storage fileset by specifying the **-gcs** option.

```
mmafmcosconfig fs1 afmfset1 --endpoint https://us-west2@storage.googleapis.com --object-fs --xattr --bucket bkt1 --mode sw --acl --gcs
```

4. Start the application inside AFM to cloud object storage.

Configuring the replication at the file system level by using the manual updates mode of the AFM to cloud object storage

Before you set up the replication, ensure that the following prerequisites are met:

- IBM Storage Scale is installed and configured on a cluster.
- A cloud account is configured with appropriate keys to create and manage buckets.
- AFM to cloud object storage is installed (**gpfs.afm.cos* rpm**) on the IBM Storage Scale cluster nodes.

The replication at the file system level can be configured by using the following methods:

1. Creating a new file system by using the **mmcrfs** command, and configuring the file system by using AFM to cloud object storage manual updates mode for the replication.

For more information, see [“Configuring a new file system for the replication by using the manual updates mode of the AFM to cloud object storage” on page 160](#) and [“An example of configuring a new file system for the replication” on page 161](#).

2. Converting an existing file system by using the **mmafmcosconfig** command, and configuring the file system by using the manual updates mode of the AFM to cloud object storage for the replication.

For more information, see [“Configuring an existing file system for the replication by using the manual updates mode of the AFM to cloud object storage” on page 164](#) and [“An example of configuring an existing file system for the replication” on page 165](#).

Configuring a new file system for the replication by using the manual updates mode of the AFM to cloud object storage

A new file system can be created and configured for replicating data by using the manual update mode of the AFM to cloud object storage.

1. Plan and provision disks to create a new file system.
 2. Create a stanza to create an NSD.
 3. Create NSDs by using these disks.
 4. Plan a storage and configure a cloud account for creating or accessing a cloud bucket, determine the secret and access keys, and create a bucket on cloud.
 5. Use the **mmafmcoskeys** to add the keys for a bucket in an IBM Storage Scale cluster.
 6. Create a file system by using the **mmcrfs** command with the **afmTarget** and **afmMode** parameters. For more information about the parameters, see the *Parameters for a new file system* section in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
 7. Use the **mmchfileset** command to configure different options mentioned in the following example.
- Note:** If you want to set a UID and a GID on a file system, use the **mmafmlocal chown \$uid:\$gid filesystempath** command.

An example of configuring a new file system for the replication

1. Create a stanza.

```
# cat nsd
%nsd:
device=/dev/sdi
nsd=nsd1
servers=Node1,Node2
usage=dataAndMetadata
failureGroup=-1
pool=system
thinDiskType=auto
%nsd:
device=/dev/sdk
nsd=nsd2
servers=Node1,Node2
usage=dataAndMetadata
failureGroup=-1
pool=system
thinDiskType=auto
%nsd:
device=/dev/sdn
nsd=nsd3
servers=Node1,Node2
usage=dataAndMetadata
failureGroup=-1
pool=system
thinDiskType=auto
%nsd:
device=/dev/sdp
nsd=nsd4
servers=Node1,Node2
usage=dataAndMetadata
failureGroup=-1
pool=system
thinDiskType=auto
```

2. Create NSDs by using the created stanza.

```
# mmcrnsd -F nsd -v no
mmcrnsd: Processing disk sdi
mmcrnsd: Processing disk sdk
mmcrnsd: Processing disk sdn
mmcrnsd: Processing disk sdp
mmcrnsd: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.

# mmlsnsd
File system      Disk name          NSD servers
-----
(free disk)    nsd1                Node1,Node2
(free disk)    nsd2                Node1,Node2
(free disk)    nsd3                Node1,Node2
(free disk)    nsd4                Node1,Node2
```

3. Create a bucket and provision for space on cloud object storage providers by using consoles, and determine access and secret keys.

A bucket, city1, is created on IBM Cloud Object Storage.

4. Add the keys by issuing the **mmafmcoskeys** command.

```
# mmafmcoskeys city1:s3.us-east.cloud-object-storage.appdomain.cloud set  
75fd1fa5c9e24aaea5093d198c0fa 61967eb3dfa2c66963bbc7d4e4725b14aa2aa10c4aff8
```

5. Create a file system, fs1, for the replication by using the city1 bucket and the MU mode.

```
# mmcrfs fs1 "nsd1;nsd2" -T /gpfs/fs1/ -p afmtarget=https://s3.us-east.cloud-object-  
storage.appdomain.cloud:443/city1 -p afmMode=manual-updates
```

The following disks of the fs1 file system are formatted on the c7f2n03 node:

```
nsd1: size 1142359 MB  
nsd2: size 1142359 MB  
Formatting file system ...  
Disks up to size 8.82 TB can be added to storage pool system.  
Creating Inode File  
71 % complete on Thu May 18 07:48:29 2023  
100 % complete on Thu May 18 07:48:31 2023  
Creating Allocation Maps  
Creating Log Files  
Clearing Inode Allocation Map  
Clearing Block Allocation Map  
Formatting Allocation Map for storage pool system  
Completed creation of file system /dev/fs1.  
mmcrfs: Propagating the cluster configuration data to all  
affected nodes. This is an asynchronous process.
```

6. Set required parameters. For more information, see the *Parameters for a new file system* section in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

```
# mmchfileset fs1 root -p afmObjectXattr=yes  
Fileset root changed.  
c7f2n02 18May07:50:16 [0] mmchfileset fs1 root -p afmObjectDirectoryObj=yes  
Fileset root changed.  
c7f2n02 18May07:50:25 [0] mmchfileset fs1 root -p afmObjectACL=yes  
Fileset root changed.
```

7. Mount the file system on all nodes.

```
# mmmount fs1 -a  
Thu May 18 07:51:27 EDT 2023: mmmount: Mounting file systems ...
```

8. List the file system attributes.

```
# mmrlsfileset fs1 root --afm -L  
Filesets in file system 'fs1':  
  
Attributes for fileset root:  
=====  
Status  
Path  
Id  
Root inode  
Parent Id  
Created  
Comment  
Inode space  
Maximum number of inodes  
Allocated inodes  
Permission change flag  
afm-associated  
Permission inherit flag  
Target  
storage.appdomain.cloud:443/city1  
Mode  
File Lookup Refresh Interval  
File Open Refresh Interval  
Dir Lookup Refresh Interval  
Dir Open Refresh Interval  
Linked  
/gpfs/fs1  
0  
3  
--  
Thu May 18 07:48:34 2023  
root fileset  
0  
2285568  
501760  
chmodAndSetacl  
Yes  
inheritAclOnly  
https://s3.us-east.cloud-object-  
storage.appdomain.cloud:443/city1  
manual-updates  
30 (default)  
30 (default)  
60 (default)  
60 (default)
```

```

Async Delay                  disable
Last pSnapId                 0
Display Home Snapshots       no
Number of Gateway Flush Threads 4
Prefetch Threshold           0 (default)
Eviction Enabled             yes (default)
IO Flags                     0x8280000
(afmObjectXattr,afmObjectDirectoryObj,afmObjectACL)
IO Flags2                    0x0 (default)

```

9. Create example data in the file system.

```

# mkdir /gpfs/fs1/dir1
c7f2n02 18May07:52:15 0] mkdir /gpfs/fs1/dir2
c7f2n02 18May07:52:22 0] echo 12345 > /gpfs/fs1/file1
c7f2n02 18May07:52:32 0] echo 12345 > /gpfs/fs1/file2
c7f2n02 18May07:52:34 0] echo 12345 > /gpfs/fs1/dir1/dfile1
c7f2n02 18May07:53:03 0] echo 12345 > /gpfs/fs1/dir2/dfile2

```

10. List the data.

```

# ls -R /gpfs/fs1
/gpfs/fs1:
dir1 dir2 file1 file2
/gpfs/fs1/dir1:
dfile1
/gpfs/fs1/dir2:
dfile2
c7f2n02 18May07:53:30 0]

```

11. Upload or replicate the data to the city1 bucket.

```

# mmafmcosctl fs1 root /gpfs/fs1/ upload --all
Queued      Failed      TotalData
                           (approx in Bytes)
 6          0            24
Object Upload successfully queued at the gateway.

```

12. Check the status of upload.

```

# mmafmctl fs1 getstate
Fileset Name   Fileset Target
Gateway Node   Queue Length Queue numExec
-----  -----
-----  -----
root          https://s3.us-east.cloud-object-storage.appdomain.cloud:443/city1 Active
c7f2n03        0            8

```

13. Check whether the data is replicated on the city1 bucket.

The following output is from the console of the cloud object storage:

```

Name      : dir1/
Date     : 0000-12-31 19:00:00 EST
Size     : 0 B
Type     : folder

Name      : dir1/dfile1
Date     : 2023-05-18 07:55:25 EDT
Size     : 6 B
ETag     : d577273ff885c3f84dadb8578bb41399
Type     : file
Metadata :
  X-Amz-Meta-Afm-Mode : 420
  X-Amz-Meta-Afm-Uid  : 0
  Content-Type        : binary/octet-stream
  X-Amz-Meta-Afm-Gid : 0
  X-Amz-Meta-Afm-Atime: 2023-05-18T11:53:03Z
  X-Amz-Meta-Afm-Mtime: 2023-05-18T11:53:03Z

Name      : dir2/
Date     : 0000-12-31 19:00:00 EST
Size     : 0 B
Type     : folder

Name      : dir2/dfile2
Date     : 2023-05-18 07:55:25 EDT

```

```

Size      : 6 B
ETag     : d577273ff885c3f84dadb8578bb41399
Type     : file
Metadata :
  X-Amz-Meta-Afm-Atime: 2023-05-18T11:53:11Z
  X-Amz-Meta-Afm-Mode : 420
  X-Amz-Meta-Afm-Gid  : 0
  X-Amz-Meta-Afm-Mtime: 2023-05-18T11:53:11Z
  X-Amz-Meta-Afm-Uid  : 0
  Content-Type        : binary/octet-stream

Name      : file1
Date     : 2023-05-18 07:55:25 EDT
Size      : 6 B
ETag     : d577273ff885c3f84dadb8578bb41399
Type     : file
Metadata :
  X-Amz-Meta-Afm-Gid  : 0
  X-Amz-Meta-Afm-Atime: 2023-05-18T11:52:32Z
  X-Amz-Meta-Afm-Mode : 420
  X-Amz-Meta-Afm-Mtime: 2023-05-18T11:52:32Z
  X-Amz-Meta-Afm-Uid  : 0
  Content-Type        : binary/octet-stream

Name      : file2
Date     : 2023-05-18 07:55:25 EDT
Size      : 6 B
ETag     : d577273ff885c3f84dadb8578bb41399
Type     : file
Metadata :
  X-Amz-Meta-Afm-Atime: 2023-05-18T11:52:34Z
  X-Amz-Meta-Afm-Gid  : 0
  Content-Type        : binary/octet-stream
  X-Amz-Meta-Afm-Mtime: 2023-05-18T11:52:34Z
  X-Amz-Meta-Afm-Mode : 420
  X-Amz-Meta-Afm-Uid  : 0

```

After the file system is created and configured, applications can create data in the new file system, and the data can be uploaded or downloaded to a cloud object storage manually or by using a policy with the **mmafmcosctl reconcile** command. For more information, see the *AFM to cloud object storage policy based upload for manual updates mode* section in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Configuring an existing file system for the replication by using the manual updates mode of the AFM to cloud object storage

To convert an existing IBM Storage Scale file system and configure it for the replication by using the AFM to cloud object storage, the **mmafmcosconfig** command is used. This command enables seamless integration between the file system and the cloud storage. By issuing the **mmafmcosconfig --convert** command, you can define the necessary settings to establish a replication relationship between the on-premises file system and the bucket on the cloud object storage.

After the conversion is initiated, the existing file system undergoes a series of steps to ensure compatibility with the manual updates mode of the AFM to cloud object storage. This mode allows manual updates to be synchronized between the file system and the cloud object storage. You can specify the frequency and granularity of the synchronization process by using the **mmafmcloud object storagectl upload , download and reconcile** command. The **mmafmcosconfig** command simplifies the process of converting an existing IBM Storage Scale file system and configuring it for the replication by using manual updates mode. It offers a streamlined approach for integrating on-premises file systems with a cloud object storage, ensuring data consistency and accessibility across environments.

1. Identify the IBM Storage Scale file system that needs to be converted.
2. Plan storage and configure a cloud account to create and access a cloud bucket. Determine the secret and access keys that are needed for authentication.
3. Create a bucket on the cloud or use the **mmafmcosconfig --new-bucket** option to create it.
4. Add the secret and access keys by using the **mmafmcoskeys** command.

5. Identify file types and file names that the cloud object storage does not support. Move these files to prevent replication issues.
6. Use the **mmafmcosconfig** command with the root fileset to convert the file system.

Note: Wherever a fileset name is needed, use the "root" fileset name.

7. Use the upload command to transfer objects from the file system to the cloud object storage.

An example of configuring an existing file system for the replication

1. Identify a file system, fs3, for conversion.

```
# ls -R /gpfs/fs3
/gpfs/fs3:
dir1 dir2 file1 file2
/gpfs/fs3/dir1:
dfile1
/gpfs/fs3/dir2:
dfile2
```

Note: For conversion of an existing file system to AFM to cloud object storage replication, the **objectFS** mode is used.

2. Add necessary keys.

```
# mmafmcoskeys city2:s3.us-east.cloud-object-storage.appdomain.cloud set
75fd1fa5c9e24aaea5093*98c025afa 61967eb***fa2cec16693bbc7d4e4725baa2aa10c4aff8
```

3. Convert the fs3 by issuing the **mmafmcosconfig** command.

```
# mmafmcosconfig fs3 root --endpoint https://s3.us-east.cloud-object-storage.appdomain.cloud
--new-bucket city2 --object-fs --mode mu --xattr --convert --acl --directory-object
mmafmcosconfig: Fileset root is not an AFM fileset.
mmafmcosconfig: Fileset root is not an AFM fileset.
Converting GPFS fileset to AFM manual-updates fileset...
Fileset root changed.
```

4. Verify that the file system is converted for the data replication from the AFM to the target bucket on the cloud object storage.

```
# mmlsfileset fs3 --afm -L
Filesets in file system 'fs3':

Attributes for fileset root:
=====
Status                               Linked
Path                                /gpfs/fs3
Id                                  0
Root inode                          3
Parent Id                           --
Created                            Fri May 19 03:29:10 2023
Comment                            root fileset
Inode space                         0
Maximum number of inodes           1142784
Allocated inodes                    500736
Permission change flag             chmodAndSetacl
afm-associated                      Yes
Permission inherit flag            inheritAclOnly
Target                             https://s3.us-east.cloud-object-
storage.appdomain.cloud:443/city2
Mode                               manual-updates
File Lookup Refresh Interval       120
File Open Refresh Interval         120
Dir Lookup Refresh Interval        120
Dir Open Refresh Interval          120
Async Delay                         disable
Last pSnapId                        0
Display Home Snapshots             no
Parallel Read Chunk Size          0
Number of Gateway Flush Threads   4
Prefetch Threshold                 0 (default)
Eviction Enabled                   yes (default)
IO Flags                           0x18280000
```

```
(afmObjectXattr,afmObjectDirectoryObj,afmObjectACL,afmMUPromoted)
IO Flags2
          0x0 (default)
```

5. Upload the files to the target bucket on the cloud object storage.

```
# mmafmcosctl fs3 root /gpfs/fs3/ upload --all
    Queued      Failed      TotalData
                           (approx in Bytes)
       6           0           24
Object Upload successfully queued at the gateway.
```

6. Verify that the files are replicated.

```
Name      : dir1/
Date     : 0000-12-31 19:00:00 EST
Size     : 0 B
Type     : folder

Name      : dir1/dfile1
Date     : 2023-05-19 03:35:51 EDT
Size     : 6 B
ETag     : d577273ff885c3f84dadb8578bb41399
Type     : file
Metadata :
  X-Amz-Meta-Afm-Atime: 2023-05-19T07:33:13Z
  X-Amz-Meta-Afm-Mtime: 2023-05-19T07:33:13Z
  X-Amz-Meta-Afm-Gid: 0
  X-Amz-Meta-Afm-Mode: 420
  X-Amz-Meta-Afm-Uid: 0
  Content-Type: binary/octet-stream

Name      : dir2/
Date     : 0000-12-31 19:00:00 EST
Size     : 0 B
Type     : folder

Name      : dir2/dfile2
Date     : 2023-05-19 03:35:51 EDT
Size     : 6 B
ETag     : d577273ff885c3f84dadb8578bb41399
Type     : file
Metadata :
  X-Amz-Meta-Afm-Gid: 0
  Content-Type: binary/octet-stream
  X-Amz-Meta-Afm-Mode: 420
  X-Amz-Meta-Afm-Atime: 2023-05-19T07:33:13Z
  X-Amz-Meta-Afm-Mtime: 2023-05-19T07:33:13Z
  X-Amz-Meta-Afm-Uid: 0

Name      : file1
Date     : 2023-05-19 03:35:51 EDT
Size     : 6 B
ETag     : d577273ff885c3f84dadb8578bb41399
Type     : file
Metadata :
  X-Amz-Meta-Afm-Mode: 420
  X-Amz-Meta-Afm-Atime: 2023-05-19T07:33:13Z
  X-Amz-Meta-Afm-Uid: 0
  X-Amz-Meta-Afm-Gid: 0
  Content-Type: binary/octet-stream
  X-Amz-Meta-Afm-Mtime: 2023-05-19T07:33:13Z

Name      : file2
Date     : 2023-05-19 03:35:51 EDT
Size     : 6 B
ETag     : d577273ff885c3f84dadb8578bb41399
Type     : file
Metadata :
  X-Amz-Meta-Afm-Gid: 0
  X-Amz-Meta-Afm-Uid: 0
  X-Amz-Meta-Afm-Mtime: 2023-05-19T07:33:13Z
  X-Amz-Meta-Afm-Atime: 2023-05-19T07:33:13Z
  Content-Type: binary/octet-stream
  X-Amz-Meta-Afm-Mode: 420
```

For more information, see the *AFM to cloud object storage policy based upload for manual updates mode* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Configuring the multi-site replication of AFM to cloud object storage

Consider the following points for the configuration:

- The multi-site replication of AFM to cloud object storage is enabled by default and is managed by using the **afmObjPIOct1** parameter. For more information, see the **mmchconfig** command in the *IBM Storage Scale: Command and Programming Reference Guide*.
- You can combine cloud object storage of different providers by using the same protocol for all. For example, any of Amazon S3 and IBM Cloud Object Storage buckets CEPH Object storage can be combined. Amazon S3 for AWS and Azure buckets cannot be combined.
- Create buckets with the same name across all chosen cloud object storage from providers before you establish AFM to cloud object storage fileset relations.
- The multi-site replication feature is supported only in the manual updates (MU) mode. In this mode, you can manage applied updates, which helps for strategic management of data replication processes.

This feature is supported on IBM Storage Scale Advanced Edition, IBM Storage Scale Data Management Edition, or IBM Storage Scale Erasure Code Edition.

- When a bucket with data exists and the AFM to cloud object storage relationship is created, then the relation is created by using AFM to cloud object storage multi-site replication data can be downloaded only from the first target and can subsequently uploaded to all other targets by using **mmafmcosctl upload --all** command. For more information, see the **mmafmcosctl** command in the *IBM Storage Scale: Command and Programming Reference Guide*.

1. Create a bucket with the same name in a respective cloud object storage of a provider.

In this example, the demomsr bucket is created on Amazon S3 and IBM Cloud Object Storage.

2. Add respective access and secret keys for the created buckets.

```
# mmafmcoskeys demomsr:s3.us-east-1.amazonaws.com set AKIAXXXXXXXXFQWFW47DX  
r9HZC+fWdgXXXXXXJSsItWrwlmg1Bibtp
```

```
# mmafmcoskeys demomsr:s3.us-east.cloud-object-storage.appdomain.cloud set  
503ddXXXXXX27bab2653167226 27ca1203XXXXX596f4819cd4d24ea3e332  
55cda27
```

3. Create a multi-target map by using the **mmafmconfig** command.

```
# mmafmconfig add demomsrmap1 --multi-target-map s3.us-east.cloud-object-  
storage.appdomain.cloud/c7f2n02,s3.us-east-1.amazonaws.com/c7f2n03
```

A sample output is as follows:

```
mmafmconfig: Command successfully completed  
mmafmconfig: Propagating the cluster configuration data to all affected nodes. This is an  
asynchronous process.
```

Note: The AWS endpoint in the map is first target. If data exists in a bucket, the endpoint can be used as the first target to download the data.

4. Create AFM to cloud object storage relation in the MU mode by using the export map.

```
# mmafmconfig add demomsrmap1 --multi-target-map s3.us-east.cloud-object-  
storage.appdomain.cloud/c7f2n02,s3.us-east-1.amazonaws.com/c7f2n03 --no-server-resolution
```

A sample output is as follows:

```
mmafmconfig: Command successfully completed  
mmafmconfig: Propagating the cluster configuration data to all affected nodes. This is an  
asynchronous process.
```

5. Create a few files.

- dd if=/dev/urandom of=/gpfs/fs1/demomsrfileset/file1 bs=256K count=4
4+0 records in
4+0 records out
1048576 bytes (1.0 MB, 1.0 MiB) copied, 0.00738335 s, 142 MB/s
- dd if=/dev/urandom of=/gpfs/fs1/demomsrfileset/file2 bs=256K count=4
4+0 records in
4+0 records out
1048576 bytes (1.0 MB, 1.0 MiB) copied, 0.00789662 s, 133 MB/s
- dd if=/dev/urandom of=/gpfs/fs1/demomsrfileset/file3 bs=256K count=4
4+0 records in
4+0 records out
1048576 bytes (1.0 MB, 1.0 MiB) copied, 0.00747471 s, 140 MB/s

6. Upload the files by using **mmafmcosctl** command.

```
# mmafmcosctl fs1 demomsrfileset /gpfs/fs1/demomsrfileset upload --all
```

A sample output is as follows:

Queued	Failed	TotalData (approx in Bytes)
3	0	3145728
Object Upload successfully queued at the gateway.		

7. Verify that the files are uploaded to targets by using any cloud client or one GUI.

- ./cloudclient ls aws/demomsr

A sample output is as follows:

```
[2024-03-11 13:19:57 EDT] 1.0MiB file1
[2024-03-11 13:19:56 EDT] 1.0MiB file2
[2024-03-11 13:19:58 EDT] 1.0MiB file3
```

- ./cloudclient ls ibm/demomsr

A sample output is as follows:

```
[2024-03-11 13:19:56 EDT] 1.0MiB file1
[2024-03-11 13:19:55 EDT] 1.0MiB file2
[2024-03-11 13:19:57 EDT] 1.0MiB file3
```

Chapter 15. Tuning for Kernel NFS backend on AFM and AFM DR

If AFM communication uses the NFSv3 protocol, for peak performance, tune the gateway NFS servers that host home exports and the gateway servers that support cache/primary filesets.

Most of these tuning parameters require at least the AFM client to be restarted. Ensure that the NFS server is not mounted. Unlink the AFM fileset or stop GPFS or IBM Storage Scale on the gateway node.

Tuning for 1 GigE networks is different from tuning 10 GigE networks. For 10 GigE, all settings need to be scaled up, but not necessarily by a factor of 10. Many of these settings are affected when a server is restarted. Therefore, each time a server restarts, the settings must be reset. The TCP buffer tuning is required for all 10 GigE links and for 1 GigE links where the value of **RTT** is greater than 0.

Tuning the gateway node on the NFS client

To tune the NFS client configuration on a gateway node, you can set the maximum number of (TCP) RPC requests. These RPC requests can be in-flight at a time by using the **sunrpc.tcp_slot_table_entries** or **/proc/sys/sunrpc/tcp_slot_table_entries** parameter.

For 1 GigE network, if the round-trip time is not set, keep the default value. You can increase the value to a number greater than 16 if the round-trip time is large. For 10 GigE network, ensure that this value is 48 or a number greater than 48 depending on the round-trip time.

When you set the **seqDiscardThreshold** parameter, it affects AFM or AFM DR as follows:

- If I/O requests are from a node that is not the gateway node, this parameter does not affect AFM or AFM DR.
- If the read request for an uncached file is on the gateway node, a higher **seqDiscardThreshold** value results in better performance. Because of this higher value, the gateway can cache more data. When the data is returned to the application, it is returned most probably from the cache or the primary cluster.

For more information about the **seqDiscardThreshold** parameter, see the **mmchconfig** command in the *IBM Storage Scale: Command and Programming Reference Guide*.

Tuning on both the NFS client (gateway) and the NFS server (the home/secondary cluster)

This topic describes the tuning on both the NFS client (gateway) and the NFS server (the home/secondary cluster).

You must set TCP values that are appropriate for the delay (buffer size = bandwidth * RTT).

For example, if your ping time is 50 ms, and the end-to-end network consists of all 100BT Ethernet and OC3 (155 Mbps), the TCP buffers must be the following: $0.05 \text{ sec} * 10 \text{ MB/sec} = 500 \text{ KB}$

If you are connected using a T1 line (1 Mbps) or less, do not change the default buffers. Faster networks usually benefit from buffer tuning.

The following parameters can also be used for tuning. A buffer size of 12194304 is provided here as an example value for a 1 GigE link with a delay of 120 ms. To set these values, set the following configurations in a file and load it with `sysctl -p` file name.

The following are example values. Initial testing is required to determine the best value for a particular system:

```
net.ipv4.tcp_rmem = 12194304 12194304 12194304  
net.ipv4.tcp_wmem = 12194304 12194304 12194304  
net.ipv4.tcp_mem = 16777216 16777216 16777216  
net.core.rmem_max = 12194304
```

```

net.core.wmem_max = 12194304
net.core.rmem_default = 12194304
net.core.wmem_default = 12194304
net.core.optmem_max = 12194304
net.core.netdev_max_backlog = 250000
net.ipv4.tcp_no_metrics_save = 1
net.ipv4.tcp_timestamps = 0
net.ipv4.tcp_sack = 1

```

Note: For TCP tuning, the **sysctl** value changes do not take effect until a new TCP connection is created, which occurs at NFS mount time. Therefore, for TCP changes, it is critical that the AFM fileset and the NFS client are unmounted and GPFS is shut down.

With Red Hat Enterprise Linux 6.1 and later, both the NFS client and the server perform TCP auto-tuning. It automatically increases the size of the TCP buffer within the specified limits through **sysctl**. If the client or the server TCP limits are too low, the TCP buffer grows for various round-trip time between the GPFS clusters. With Red Hat Enterprise Linux 6.1 and earlier, NFS is limited in its ability to tune the TCP connection. Therefore, do not use a version earlier than Red Hat Enterprise Linux 6.1 in the cache/primary cluster.

As a GPFS cluster might be handling local and remote NFS clients, you can set the GPFS server values for the largest expected round-trip time of any NFS client. This ensures that the GPFS server can handle clients at various locations. Then, on the NFS clients, set the TCP buffer values that are appropriate for the GPFS cluster that they are accessing.

The gateway node is both an NFS server for standard NFS clients if they exist and an NFS client for communication with the home/secondary cluster. Ensure that the TCP values are set appropriately because values that are either too high or too low can negatively impact performance.

If performance continues to be an issue, increase the buffer value by up to 50%. If you increase the buffer value by more than 50%, it might have a negative effect.

Tuning the NFS server on the home/secondary cluster or the NFS server

This topic describes the tuning on the NFS server on the home/secondary cluster or the NFS server.

Table 11. NFS server parameters	
Parameter name	Description
/proc/fs/nfsd/max_block_size	Set to 1 MB for improved performance.
/proc/fs/nfsd/threads	Set to a minimum value of 32. You can set this value to greater than 128 depending on the throughput capacity and the round-trip time between the cache/primary and home/secondary clusters. Determining the correct value might take a few trials.

Table 11. NFS server parameters (continued)

Parameter name	Description
nfsPrefetchStrategy	<p>Set it to a number between 5–10. As AFM uses NFS, ensure that this parameter is set on the home/secondary GPFS cluster.</p> <p>After the NFS values are set, you can mount and access the AFM filesets. The first time the fileset is accessed the AFM NFS client mounts the home/secondary server or servers. To see these mounts on a gateway node, enter the following command: cat /proc/mounts.</p> <p>The system displays the mount point and the mount options. If the wsize and rsize values are not 1 MB, you can adjust the parameters and mount the AFM filesets again to get the correct values.</p> <p>For more information about the nfsPrefetchStrategy parameter, see the mmchconfig command in the <i>IBM Storage Scale: Command and Programming Reference Guide</i>.</p>

Chapter 16. Configuring call home

The call home must be configured before it can be used. The following topics describe various ways to configure the IBM Storage Scale call home, and provide several configuration examples.

Configuring call home to enable manual and automated data upload

The call home component needs to be configured before it can be used to perform manual and automated data uploads.

The configuration process consists of the following steps:

1. Configure the call home settings.
2. Create call home groups.

After performing these steps, you will be able to use the `mmcallhome run SendFile --file file [--desc DESC | --pmr {xxxxx.yyy.zzz | TSxxxxxxxx}]` command to upload a specific file to the ECuRep. Any data collection schedules that are configured are run regularly.

Configuring call home settings

To configure call home settings, perform the following steps:

1. If you are using proxy, configure the proxy settings:
 - a. Set the proxy location and authentication settings, by using the following command:

```
mmcallhome proxy change --proxy-location ProxyLocation  
--proxy-port ProxyPort [--proxy-username ProxyUsername  
--proxy-password ProxyPassword]
```

- b. Enable the proxy by using the following command:

```
mmcallhome proxy enable [--with-proxy-auth]
```

- 2. Set up the customer information by using the following command:

```
mmcallhome info change --customer-name CustomerName  
--customer-id CustomerId --email Email --country-code CountryCode
```

Note:

In special cases, the following customer IDs must be used:

- For developer edition: DEVLIC
- For test edition for customers who are trying IBM Storage Scale: TRYBUY

The country code must correspond to the contact country. The contact country information can be found in the same source where the customer ID is found. Since the customer number is not unique throughout all countries, a country code is also required to unambiguously identify a customer. The country code must be in the ISO 3166-1 alpha-2 format. For example, US for USA, DE for Germany. For more details, see [ISO 3166-1 alpha-2](#).

- 3. Set up the scheduled data collection if needed, by using the following commands:

```
mmcallhome schedule add --task DAILY  
mmcallhome schedule add --task WEEKLY
```

Note: Explicit enabling is only needed if call home is running on nodes that have IBM Storage Scale version 5.0.1 or older installed.

4. Enable the call home capability by using the following command:

```
mmcallhome capability enable
```

Creating call home groups

There are two ways to create call home groups: automatically and manually.

Automatic group creation allows users to create homogeneous groups, assigning all compatible cluster nodes to one of the groups. Automatic groups are created by using the **mmcallhome group auto** command. For more information regarding the automatic group creation, see [“Configuring the call home groups automatically” on page 174](#).

Manual group creation allows the users to fine-tune the contents of the call home groups. For example, you can make the following changes to the call home groups:

- Grouping specific nodes within a group.
- Create call home groups with inhomogeneous sizes.
- Change the contents of an existing group without influencing other groups.

Manual group is created by using the **mmcallhome group add** command. For more information about the automatic group creation, see [“Configuring the call home groups manually” on page 174](#).

Configuring the call home groups manually

Manual group creation is meant for specific use cases, where you need to customize the contents for the new groups.

It is performed by using the following command:

```
mmcallhome group add GroupName server  
[--node {all | ChildNode[,ChildNode...]}]
```

This assigns the nodes that are specified by using the --node option and the node that is specified as server to the new call home group named GroupName. The server node is set to be the call home node of the created group.

All group nodes must have call home packages that are installed, and the call home node must be able to connect to esupport.ibm.com. If the proxy call home settings are enabled, then a proxy is used to test the connectivity. Otherwise, a direct connection is attempted.

If the call home node has a release version of 5.0.1.0 or later, then the new group is automatically set to track the global call home settings.

Note: If you are using a mixed cluster setup and specify a node with a release version earlier than 5.0.1.0 as the call home node, then the created group has a group-specific configuration with default values, and must be manually configured. In such cases, run the same commands that you ran before to configure global call home settings from one of the nodes of the newly created groups. For more information, see the *Configuring call home settings* section in [“Configuring call home to enable manual and automated data upload” on page 173](#).

Configuring the call home groups automatically

If you want to distribute all compatible cluster nodes into call home groups automatically and create homogeneous groups, you must use the **mmcallhome group auto** command after the call home settings are configured.

All compatible nodes that have call home packages and are not yet a part of any call home groups can be redistributed into new call home groups by using the **mmcallhome group auto** command. All the newly created call home groups then use the global call home configuration. The following actions are run in this process:

1. The nodes that can access esupport.ibm.com and have call home packages that are installed on them are detected.

Note: If a proxy is specified by the **mmcallhome proxy change** command and enabled by the **mmcallhome proxy enable** command, then the specified proxy is used for detecting the nodes that have access to esupport.ibm.com. If the proxy configuration is disabled, direct connections are attempted instead.

2. A minimal subset of these nodes is selected, so that all nodes, which are supposed to be distributed into groups, can be distributed into groups with a maximum recommended size of 1024 nodes.
3. New groups are created and set to use the global call home settings.

If you want to redistribute all nodes, which are currently assigned to any groups, use the **--force** option as shown. The use of the **--force** option effectively deletes all current groups before creating new ones.

```
mmcallhome group auto --force
```

If you want to manually specify the call home nodes to use for the new groups, you can use the **--server** option as shown:

```
mmcallhome group auto [--server {ServerName[,ServerName...]}]
```

In such cases, the following rules apply:

- The number of groups that are created is the same as the number of the specified call home nodes.
- The access to esupport.ibm.com is not checked for any call home nodes.
- Each group gets one of the specified call home nodes that are assigned to it.
- All compatible nodes are distributed between these groups.

If you want to distribute only a part of your cluster into the call home groups, you can use the **--nodes** option:

```
mmcallhome group auto --nodes {all | ChildNode1[,ChildNode2...]}
```

Configuring call home using GUI

The call home feature provides a communication channel that automatically notifies the IBM service personnel about the issues reported in the system. You can also manually upload diagnostic data files and associate them with a PMR through the GUI.

You can use the **Call Home** page in the GUI to perform the following tasks:

- Enable call home feature on the cluster.
- Select one or more call home nodes that share the data with the IBM Support.
- Specify the contact information to be used by the IBM Support if there are any issues.
- Specify the proxy information that is needed to create a communication channel between the call home nodes and IBM support.
- Test connection with the IBM server.

Collecting data and sharing it with IBM Support

The call home shares support information and your contact information with IBM on a schedule basis. The IBM Support monitors the details that are shared by the call home and takes necessary action in case of any issues or potential issues. Enabling call home reduces the response time for the IBM Support to address the issues. The call home automatically shares data with the IBM support based on a schedule. The GUI does not support to change the data gathering and sharing schedules.

You can also manually upload the diagnostic data that is collected through the **Settings > Diagnostic Data** page in the GUI to share the diagnostic data to resolve a Problem Management Record (PMR). To upload data manually, perform the following steps:

1. Go to **Support > Diagnostic Data**.
2. Collect diagnostic data based on the requirement. You can also use the previously collected data for the upload.
3. Select the relevant data set from the **Previously Collected Diagnostic Data** section and then right-click and select **Upload to PMR**.
4. Select the PMR to which the data must be uploaded and then click **Upload**.

Call home configuration examples

The following section gives some examples of the call home configuration.

Until IBM Storage Scale 4.2.3.7, each call home group had its own configuration, which had to be configured and managed separately. All call home nodes with the release version between 4.2.3.7 to 5.0.0.x are set to use global call home settings, after the first change of the corresponding setting from the default value. The change of values happens automatically if the groups are created using the **mmcallhome group auto** command.

For all call home nodes with IBM Storage Scale version 5.0.1.0 or later, the call home nodes are automatically set to use the global call home configuration upon their creation.

For the following use cases we assume the following customer information:

- Customer name: User1
- Customer ID: 123456
- E-mail: customer@ibm.com
- Country-code: JP

Use Case 1: To automatically create call home groups for all Linux nodes in the cluster where call home packages are installed, and enable all call home features.

Note: For this use case, we assume the following:

- Call home has not been configured before.
- Some of the nodes have a direct connectivity to esupport.ibm.com.
- Automatic daily and weekly data collection is to be enabled.
- The **mmcallhome** command is run from a node which has IBM Storage Scale version 5.0.2. or later.

1. Set the customer information:

```
[root@g5001-21 ~]# mmcallhome info change --customer-name
User1 --customer-id 123456 --email customer@ibm.com --country-code JP
Call home country-code has been set to JP
Call home customer-name has been set to User1
Call home customer-id has been set to 123456
Call home email has been set to customer@ibm.com
```

2. Enable call home to actually send data:

```
[root@g5001-21 ~]# mmcallhome capability enable
By accepting this request, you agree to allow IBM
and its subsidiaries to store and use your contact information
and your support information anywhere they do business worldwide.
For more information, please refer to the Program license
agreement and documentation. If you agree, please respond
with "accept" for acceptance, else with "not accepted" to decline.
(accept / not accepted)
accept
Call home enabled has been set to true

Additional messages:
License was accepted. Callhome enabled.
```

3. Create the call home groups automatically:

```
[root@g5020-31 ~]# mmcallhome group auto
[I] Analysing the cluster...
[I] Creating <1> new call home groups:
[I] Call home child nodes = g5020-31.localnet.com,g5020-32.localnet.com,g5020-34.localnet.com
[I] Call home group autoGroup_1 has been created
[I] Nodes without call home: <1> (g5020-33.localnet.com)
[I] Updating call home node classes...
g5020-32.localnet.com: QOS configuration has been installed and broadcast to all nodes.
g5020-32.localnet.com: QOS configuration has been installed and broadcast to all nodes.
[I] The automatic group creation completed successfully.
```

Use Case 2: To distribute all Linux nodes in the cluster where call home packages are installed into two call home groups, and set the nodes g5001-21 and g5001-22 as their call home nodes.

Note: For this use case, we assume the following:

- Call home has not been configured before.
- Both the call home nodes require an authenticated proxy.
- Enable only weekly data collection.
- The **mmcallhome** command is run from a node which has IBM Storage Scale version 5.0.2. or later.

1. Set the customer information:

```
[root@g5001-21 ~]# mmcallhome info change --customer-name
User1 --customer-id 123456 --email customer@ibm.com --country-code JP
Call home country-code has been set to JP
Call home customer-name has been set to User1
Call home customer-id has been set to 123456
Call home email has been set to customer@ibm.com
```

2. Disable the daily schedule and event-based uploads:

```
[root@g5050-11 ~]# mmcallhome schedule delete --task DAILY
Call home daily has been set to disabled
[root@g5050-11 ~]# mmcallhome schedule delete --task EVENT
Call home event has been set to disabled
```

3. Define the proxy settings and enable proxy:

```
[root@g5001-21 ~]# mmcallhome proxy change
--proxy-location 192.168.0.10 --proxy-port 5085
--proxy-username clusteradmin --proxy-password MyPass
Call home proxy-port has been set to 5085
Call home proxy-username has been set to clusteradmin
Call home proxy-password has been set to MyPass
Call home proxy-location has been set to 192.168.0.10
[root@g5001-21 ~]# mmcallhome proxy enable --with-proxy-auth
Call home proxy-enabled has been set to true
Call home proxy-auth-enabled has been set to true
```

4. Enable call home to send data:

```
[root@g5001-21 ~]# mmcallhome capability enable
By accepting this request, you agree to allow
IBM and its subsidiaries to store and use your
contact information and your support information
anywhere they do business worldwide. For more
information, please refer to the Program license
agreement and documentation. If you agree, please
respond with "accept" for acceptance, else with
"not accepted" to decline.
(accept / not accepted)
accept
Call home enabled has been set to true
```

```
Additional messages:  
License was accepted. Callhome enabled.
```

5. Create the call home groups automatically, while specifying the call home nodes:

```
[root@g5020-31 ~]# mmcallhome group auto --server g5020-31,g5020-32  
[I] Analysing the cluster...  
[I] Creating <2> new call home groups:  
[I] Call home child nodes = g5020-31.localnet.com,g5020-34.localnet.com  
[I] Call home group autoGroup_1 has been created  
[I] Call home child nodes = g5020-32.localnet.com  
[I] Call home group autoGroup_2 has been created  
[I] Nodes without call home: <1> (g5020-33.localnet.com)  
[I] Updating call home node classes...  
QOS configuration has been installed and broadcast to all nodes.  
QOS configuration has been installed and broadcast to all nodes.  
[I] The automatic group creation completed successfully.
```

Use Case 3: To automatically create call home groups for all Linux nodes in the cluster where call home packages are installed, but disable the scheduled data collection.

Note: For this use case, we assume the following:

- Call home has been configured before, but must be reconfigured. Ensure that the old settings are removed, and the old groups are deleted.
- Both the call home nodes require an authenticated proxy.
- Neither weekly nor daily nor event-based data collection must be enabled, as data upload is only done on demand. For example, PMRs.
- The **mmcallhome** command is run from a node which has IBM Storage Scale version 5.0.2. or later.

1. Set the customer information:

```
[root@g5001-21 ~]# mmcallhome info change --customer-name  
User1 --customer-id 123456 --email customer@ibm.com --country-code JP  
Call home country-code has been set to JP  
Call home customer-name has been set to User1  
Call home customer-id has been set to 123456  
Call home email has been set to customer@ibm.com
```

2. Disable the proxy configuration:

```
[root@g5001-21 ~]# mmcallhome proxy disable  
Call home proxy-enabled has been set to false  
Call home proxy-auth-enabled has been set to false
```

3. Disable all upload schedules and event-based uploads:

```
[root@g5050-11 ~]# mmcallhome schedule delete --task EVENT  
Call home event has been set to disabled
```

4. Enable call home to send data when needed:

```
[root@g5001-21 ~]# mmcallhome capability enable  
By accepting this request, you agree to allow IBM and its  
subsidiaries to store and use your contact information  
and your support information anywhere they do business worldwide.  
For more information, please refer to the Program license agreement  
and documentation. If you agree, please respond with "accept" for  
acceptance, else with "not accepted" to decline.  
(accept / not accepted)  
accept  
Call home enabled has been set to true  
  
Additional messages:  
License was accepted. Callhome enabled.
```

5. Create the call home groups automatically, while removing all previously existing groups:

```
[root@g5020-31 ~]# mmcallhome group auto --force
[I] Analysing the cluster...
[I] Deleting old call home groups (--force mode)
[I] Creating <1> new call home groups:
[I] Call home child nodes = g5020-31.localnet.com,g5020-32.localnet.com,g5020-34.localnet.com
[I] Call home group autoGroup_1 has been created
[I] Nodes without call home: <1> (g5020-33.localnet.com)
[I] Updating call home node classes...
QOS configuration has been installed and broadcast to all nodes.
QOS configuration has been installed and broadcast to all nodes.
[I] The automatic group creation completed successfully.
```

Use cases for detecting system changes by using the **mmcallhome** command

The topic describes the use cases for the **mmcallhome** command for detecting system changes:

Consider the following assumptions for all use cases:

- Call home is configured.
- At least two daily or weekly data uploads were already completed.
- A file system **mari** exists in the system.

Note: Only the configuration data that is collected can be compared. Since the configuration data for all the objects in IBM Storage Scale are not collected by call home, the configuration data for these objects is not listed in the configuration difference.

Use Case 1: Check the changes that happened on the system between the last two data collections that were uploaded to IBM support

1. Run the following command to check whether any changes are detected:

```
[root@mari-11 ~]# mmcallhome status diff
There are no differences.
```

2. Run the following command to unmount **mari**:

```
# mmunmount mari -a -f
# mmdelfs mari
# mmcallhome run GatherSend --task DAILY
```

3. Run the following command to check whether any changes happened to the system after the unmount:

```
[root@mari-11 ~]# mmcallhome status diff
```

The system gives an output similar to the following:

```
Fs Data
Device Name = mari      (deleted)

Nsd Data
Nsd Id  = 0A00640B5E79DB35      (modified)
Fs Name : mari  --> (free disk)

Nsd Id  = 0A00640B5E79DB46      (modified)
Fs Name : mari  --> (free disk)
```

Note: You can use the following command to generate a more detailed output:

```
[root@mari-11 ~]# mmcallhome status diff -v
```

- The system gives an output similar to the following:

```
Fs Data

Device Name = mari      (deleted)
Rw Options   : rw
Mount Options : rw,atime,mtime,userquota;groupquota;filesetquota,nfssync,nodev
Drive Letter : N/A
Automount Option : yes
Perfileset Quota : no
Mount Point   : /mnt/mari
Device Name   : mari
Quota Option  : userquota;groupquota;filesetquota
Mtime Option  : mtime
Fs Type       : local
Dmapi Enabled : no
Atime Option  : atime
Maintenance Mode : no
Device Minor Number : 152
Other Mount Options : nfssync,nodev
Owning Cluster Name : gpfs-cluster-1.localnet.com
Remote Device Name : mari

Nsd Data

Nsd Id = 0A00640B5E79DB35      (modified)
Fs Name : mari --> (free disk)

Nsd Id = 0A00640B5E79DB46      (modified)
Fs Name : mari --> (free disk)
```

- If just one configuration option in the system was changed, then the system gives the following output:

```
Nodeclass Data

Nodeclass = GUI_MGMT_SERVERS      (modified)
Allmembers : mari-12.localnet.com --> mari-12.localnet.com,mari-13.localnet.com
Membernodes : mari-12.localnet.com --> mari-12.localnet.com,mari-13.localnet.com
```

Use Case 2: Check the changes that happened on the system between a selected historic data collection and the most recent one that is uploaded to IBM support

Note: For this use case, consider that the history of the data collections that are kept on the system is limited. If a specified number is reached for the data collection, then the old files are deleted. Hence, the old files are no longer available for configuration comparisons.

- Run the following command to select a previous data collection to compare with the most recent data collection:

```
[root@mari-11 ~]# mmcallhome status list -v --task weekly
```

The system gives an output similar to the following:

```
autoGroup_1 weekly 20200324122356.621 20200324122420 success RC=0          /tmp/mmfs/callhome/rsENUploaded/
31790849437793.5_0_5_0.123456.DE.ibmtest.autoGroup_1.gat_weekly.g_weekly.scale.2020032412235621.c10.DC

autoGroup_1 weekly 20200329030501.922 20200329030529 success RC=0          /tmp/mmfs/callhome/rsENUploaded/
31790849437793.5_0_5_0.123456.DE.ibmtest.autoGroup_1.gat_weekly.g_weekly.scale.20200329030501922.c10.DC

autoGroup_1 weekly 20200401112254.712 20200401112322 success RC=0          /tmp/mmfs/callhome/rsENUploaded/
31790849437793.5_0_5_0.123456.DE.ibmtest.autoGroup_1.gat_weekly.g_weekly.scale.20200401112254712.c10.DC

autoGroup_1 weekly 20200401121836.059 20200401121844 success RC=0          /tmp/mmfs/callhome/rsENUploaded/
31790849437793.5_0_5_0.123456.DE.ibmtest.autoGroup_1.gat_weekly.g_weekly.scale.20200401121816059.c10.DC

autoGroup_1 weekly 20200405030502.039 20200405030527 success RC=0          /tmp/mmfs/callhome/rsENUploaded/
31790849437793.5_0_5_0.123456.DE.ibmtest.autoGroup_1.gat_weekly.g_weekly.scale.20200405030502039.c10.DC

autoGroup_1 weekly 20200412030501.677 20200412030526 success RC=0          /tmp/mmfs/callhome/rsENUploaded/
31790849437793.5_0_5_0.123456.DE.ibmtest.autoGroup_1.gat_weekly.g_weekly.scale.20200412030501677.c10.DC
```

Note: Usually all historic data collections that are gathered on the system can be used for the comparison. However, in this case only the weekly data collections is requested.

2. Run the following command to select one data collection to compare with the most recent data collection:

```
[root@mari-11 ~]# mmcallehome status diff --old 20200412
```

The system gives an output similar to the following:

```
Nodeclass Data  
Nodeclass = GUI_MGMT_SERVERS (modified)  
Allmembers : mari-12.localnet.com --> mari-12.localnet.com,mari-13.localnet.com  
Membernodes : mari-12.localnet.com --> mari-12.localnet.com,mari-13.localnet.com
```

Note: If you select a file that was already removed, the system throws an error. In this case, choose another data collection.

3. Run the following command to compare a data collection that was created 3 days back:

```
[root@mari-11 ~]# mmcallehome status diff -last-days 3
```

The system gives an output similar to the following:

```
Nodeclass Data  
Nodeclass = GUI_MGMT_SERVERS (modified)  
Allmembers : mari-12.localnet.com --> mari-12.localnet.com,mari-13.localnet.com  
Membernodes : mari-12.localnet.com --> mari-12.localnet.com,mari-13.localnet.com
```

Use Case 3: Check the changes that happened on the system between two distinct historic data collections

Run the following command to select and compare two distinct data collections:

```
[root@mari-11 ~]# mmcallehome status diff --old 20200412 --new 20200416 -v
```

The system gives an output similar to the following:

```
Fs Data  
Device Name = mari (created)  
Rw Options : rw  
Mount Options : rw,atime,mtime,userquota;groupquota;filesetquota,nfssync,nodev  
Drive Letter : N/A  
Automount Option : yes  
Perfileset Quota : no  
Mount Point : /mnt/mari  
Device Name : mari  
Quota Option : userquota;groupquota;filesetquota  
Mtime Option : mtime  
Fs Type : local  
Dmapi Enabled : no  
Atime Option : atime  
Maintenance Mode : no  
Device Minor Number : 152  
Other Mount Options : nfssync,nodev  
Owning Cluster Name : gpfs-cluster-1.localnet.com  
Remote Device Name : mari  
  
Nsd Data  
Nsd Id = 0A00640B5E79DB35 (modified)  
Fs Name : (free disk) --> mari  
Nsd Id = 0A00640B5E79DB46 (modified)  
Fs Name : (free disk) --> mari
```

Chapter 17. Integrating IBM Storage Scale Cinder driver with Red Hat OpenStack Platform 16.1

Integrate the IBM Storage Scale file system Cinder driver with Red Hat® OpenStack Platform (RHOSP) 16.1.3 as follows.

The following are the prerequisites before you start integrating IBM Storage Scale Cinder driver with RHOSP:

- OpenStack Provider Network: 192.168.24.x/24
- IBM Storage Scale CES Network: 10.0.0.x/24
- All the overcloud nodes have access to IBM Storage Scale CES Network.

The IBM Storage Scale volume driver, named `gpfs.py`, enables the use of IBM Storage Scale in a fashion similar to that of the NFS driver. With the IBM Storage Scale driver, instances do not access a storage device at the block level. Instead, volume backing files are created in the IBM Storage Scale file system and mapped to instances, which emulate a block device. The Image service can also be configured to store glance images in the IBM Storage Scale file system.

In a scenario, when both compute and controller nodes are not running IBM Storage Scale software and do not have access to IBM Storage Scale file system directly as a local file system, the NFS export is created on the volume path and make it available on the controller and compute nodes. This deployment mode is usually referred to as Remote IBM Storage Scale Access. For more information, see [IBM Storage Scale volume driver - Mode 3](#).

Configuring IBM Storage Scale cluster to enable Cinder driver with RHOSP

Before deploying the Cinder backend configuration in RHOSP, configure the IBM Storage Scale cluster to enable Cinder driver with RHOSP as follows.

1. Enable CES NFS service in IBM Storage Scale.

```
# mmcse service list  
Enabled services: NFS  
NFS is running
```

2. Create an independent fileset to store OpenStack components (Cinder, Glance) data.

```
# mmcrfileset fs1 openstack --inode-space new  
# mmrlinkfileset fs1 openstack -J /ibm/fs1/openstack  
# mmrlsfileset fs1 openstack  
Filesets in file system 'fs1':  
Name      Status      Path  
openstack  Linked      /ibm/fs1/openstack
```

3. Create following directories for OpenStack components:

- Cinder

```
# mkdir -p /ibm/fs1/openstack/cinder/volumes
```

- Glance

```
# mkdir -p /ibm/fs1/openstack/glance/images
```

4. Create an NFS export for the following directories:

```
# mnfs export add /ibm/fs1/openstack/cinder/volumes --client  
"10.0.0.0/24(Access_Type=RW,SQUASH=no_root_squash)"
```

```
# mmnfs export add /ibm/fs1/openstack/glance/images --client  
"10.0.0.0/24(Access_Type=RW,SQUASH=no_root_squash)"
```

5. Update the NFS configuration to support NFS v4.1.

```
# mmnfs config change MINOR_VERSIONS=1  
mmnfs: The NFS configuration was changed successfully.  
mmnfs: NFS server restarted on all NFS nodes on which NFS server is running.
```

Note: If there are multiple protocol nodes serving the NFS share, ensure the following requirements:

- **ssh_host_keys** on all the protocol nodes are identical before you integrate IBM Storage Scale Cinder driver with RHOSP. One way to do that is by copying **ssh_host_keys** from one protocol node to all other protocol nodes in the /etc/ssh directory.
- **recover_lost_locks** module/kernel parameter is enabled on all the compute nodes. To enable **recover_lost_locks** on the compute nodes, issue the following commands:

```
# cat > /etc/modprobe.d/nfs4-locks.conf <<EOF  
options nfs recover_lost_locks=1  
EOF
```

```
# [ -d "/sys/module/nfs" ] && echo Y > /sys/module/nfs/parameters/recover_lost_locks
```

Deploying IBM Storage Scale Cinder backend configuration in RHOSP

Use the following RHOSP deployment procedure for remote IBM Storage Scale access mode:

1. Prepare the Triple-O IBM Storage Scale environment file.

- Create three YAML templates that represent a set of plans to integrate IBM Storage Scale Cinder driver (mode 3) in RHOSP:

cinder-spectrumscale-config.yaml

This environment file consists of all the configuration parameters that are needed to integrate the IBM Storage Scale driver with RHOSP. For more information, see “[Sample cinder-spectrumscale-config.yaml file](#)” on page 186.

spectrumscale-pre-config.yaml

This heat template ensures that a cinder-volume path is bind mounted on all the overcloud nodes. For more information, see “[Sample spectrumscale-pre-config.yaml file](#)” on page 187.

spectrumscale-post-config.yaml

This heat template ensures that a cinder-volume path is mounted with NFS share on all the overcloud nodes. For more information, see [Sample spectrumscale-post-config.yaml file](#).

- Keep these YAML templates available in the home directory of stack user of director node (that is, /home/stack/).

2. Add **cinder-spectrumscale-config.yaml** template as an environment file to integrate IBM Storage Scale Cinder driver for final deployment. Use the **-e** option in deployment command to include the **cinder-spectrumscale-config.yaml** environment file.

```
openstack overcloud deploy --templates \  
-e /home/stack/templates/node-info.yaml \  
-e /home/stack/containers-prepare-parameter.yaml \  
-e /home/stack/cinder-spectrumscale-config.yaml
```

For more information, see [Configuring a basic Overcloud with CLI tools](#).

Limitations of integrating IBM Storage Scale Cinder driver with RHOSP

Limitations encountered when you integrate IBM Storage Scale Cinder driver with RHOSP:

- The IBM Storage Scale Cinder driver accepts single CES IP as **nas_host** parameter value in order to mount the cinder-volume filesset through Triple-O integration.
- NFS v3 is not supported as a value for **GlanceNfsOptions** or **SpectrumScaleNFSSOptions** parameters.
- Usage of copy-on-write feature to create volume from an image is not supported.
- If either **nas_host** or **gpfs_mount_point_base** changes after the initial deployment, you need to remove the older NFS share entry from /etc/fstab from all the overcloud nodes.
- The following features are not supported by IBM Storage Scale cinder driver in RHOSP:
 - Multi attach volume
 - Create a consistency group from a consistency group snapshot
 - Create a consistency group from a consistency group
 - Extend attached volume
 - Backup volume

Triple-O heat template environment parameters

List of heat template environment parameters.

Triple-O Heat Template Environment Parameters

The following table lists the Triple-O heat template environment parameters.

Table 12. Triple-O Heat Template Environment Parameters		
Configuration Option	Type	Description
GlanceBackend	String	Define the backend to use for glance. Set to file to use file-based storage for images. The overcloud saves these files in a mounted NFS share for glance.
GlanceNfsEnabled	Boolean	When GlanceBackend is set to file, GlanceNfsEnabled enables images to be stored through NFS in a shared location so that all Controller nodes have access to the images. If disabled, the overcloud stores images in the file system of the Controller node. Set to true.
GlanceNfsShare	String	The NFS share to mount for image storage.
GlanceNfsOptions	String	The NFS mount options for image storage.
SpectrumScaleNFSServer	String	Triple-O Heat template environment parameter to define IBM Storage Scale CES IP for NFS service. This value must match with nas_host from cinder configuration parameters.
SpectrumScaleNFSSOptions	String	Triple-O Heat template environment parameter to define NFS mount option on all overcloud nodes.

Table 12. Triple-O Heat Template Environment Parameters (continued)

Configuration Option	Type	Description
SpectrumScaleVolumeShare	String	Triple-O Heat template environment parameter to define NFS share Volume path. This value must match with gpfs_mount_point_base from cinder configuration parameters. For example, /ibm/fs1/openstack/cinder/volumes
CinderVolumeOptVolumes	String	Enables custom configuration files on the host to be made available to the cinder-volume service when it's running in a container. This value must match with gpfs_mount_point_base from cinder configuration parameters. For example, /ibm/fs1/openstack/cinder/volumes
NovaLibvirtOptVolumes	String	Enables custom configuration files on the host to be made available to the nova-libvirt service when it's running in a container. This value must match with gpfs_mount_point_base from cinder configuration parameters. For example, /ibm/fs1/openstack/cinder/volumes
CinderEnableIscsiBackend	Boolean	Enables the iSCSI backend. Set to false.

Sample IBM Storage Scale Cinder configuration YAML file

Sample YAML file that displays IBM Storage Scale Cinder configuration file data.

Sample `cinder-spectrumscale-config.yaml` file

```

resource_registry:
  # Path of the IBM Spectrum Scale Pre-config file to mount the NFS Share
  # required for Cinder driver on all overcloud nodes
  OS::TripleO::NodeExtraConfig: /home/stack/spectrumscale-pre-config.yaml
  OS::TripleO::NodeExtraConfigPost: /home/stack/spectrumscale-post-config.yaml

parameter_defaults:
  CinderEnableIscsiBackend: false
  GlanceBackend: file
  GlanceNfsEnabled: true
  GlanceNfsShare: 10.0.0.101:/ibm/fs1/openstack/glance/images
  GlanceNfsOptions:
    rw,vers=4.1,sync,bg,_netdev,intr,context=system_u:object_r:container_file_t:s0
    SpectrumScaleNFSServer: 10.0.0.101
    SpectrumScaleCinderNFSOptions:
      rw,vers=4.1,sync,bg,_netdev,context=system_u:object_r:container_file_t:s0
      SpectrumScaleNovaNFSOptions: rw,vers=4.1,sync,bg,_netdev,context=system_u:object_r:nfs_t:s0
      SpectrumScaleVolumeShare: /ibm/fs1/openstack/cinder/volumes
      CinderVolumeOptVolumes:
        - /ibm/fs1/openstack/cinder/volumes:/ibm/fs1/openstack/cinder/volumes:slave
      NovaLibvirtOptVolumes:
        - /ibm/fs1/openstack/cinder/volumes:/ibm/fs1/openstack/cinder/volumes:slave
  ExtraConfig:
    cinder::config::cinder_config:
      tripleo_spectrumscale/volume_driver:
        value: cinder.volume.drivers.ibm_gpfs.GPFSNFSDriver
      tripleo_spectrumscale/gpfs_mount_point_base:
        value: /ibm/fs1/openstack/cinder/volumes
      tripleo_spectrumscale/volume_backend_name:
        value: tripleo_spectrumscale
      tripleo_spectrumscale/nas_host:
        value: 10.0.0.101
      tripleo_spectrumscale/nas_login:
        value: root
      tripleo_spectrumscale/nas_password:
        value: abc

```

```

tripleo_spectrumscale/nas_share_path:
  value: /ibm/fs1/openstack/cinder/volumes
tripleo_spectrumscale/nfs_mount_point_base:
  value: /ibm/fs1/openstack/cinder/volumes
cinder_user_enabled_backends: ['tripleo_spectrumscale']

```

Sample spectrumscale-pre-config.yaml file

```

heat_template_version: rocky

description: >
  Cinder host pre configuration for IBM SpectrumScale Mode 3

parameters:
  server:
    type: string
  SpectrumScaleNFSServer:
    type: string
  SpectrumScaleVolumeShare:
    type: string
  DeployIdentifier:
    type: string

resources:
  CustomExtraConfigPre:
    type: OS::Heat::SoftwareConfig
    properties:
      group: script
      config:
        str_replace:
          template: |
            #!/bin/bash
            GPFS_VOLUME_PATH=GPFS_VOLUME_SHARE/`echo -n GPFS_NFS_SERVER:GPFS_VOLUME_SHARE | md5sum | awk '{print $1; exit}'`
            mkdir -p $GPFS_VOLUME_PATH
      params:
        GPFS_VOLUME_SHARE: {get_param: SpectrumScaleVolumeShare}
        GPFS_NFS_SERVER: {get_param: SpectrumScaleNFSServer}
  CustomExtraDeploymentPre:
    type: OS::Heat::SoftwareDeployment
    properties:
      server: {get_param: server}
      config: {get_resource: CustomExtraConfigPre}
      actions: ['CREATE','UPDATE']
      input_values:
        deploy_identifier: {get_param: DeployIdentifier}

outputs:
  deploy_stdout:
    description: Pre-configuration script for IBM SpectrumScale Mode 3
    value: {get_attr: [CustomExtraDeploymentPre, deploy_stdout]}

```

Sample spectrumscale-post-config.yaml file

```

heat_template_version: rocky

description: >
  Cinder host post configuration for IBM SpectrumScale Mode 3

parameters:
  servers:
    type: json
  SpectrumScaleNFSServer:
    type: string
  SpectrumScaleVolumeShare:
    type: string
  SpectrumScaleCinderNFSOptions:
    type: string
  SpectrumScaleNovaNFSOptions:
    type: string
  DeployIdentifier:
    type: string
  EndpointMap:
    default: {}
    type: json

```

```

resources:
  CustomExtraConfig:
    type: OS::Heat::SoftwareConfig
    properties:
      group: script
      config:
        str_replace:
          template: |
            #!/bin/bash
            if ! grep -q 'GPFS_NFS_SERVER:GPFS_VOLUME_SHARE' /etc/fstab
            then
              if hiera -c /etc/puppet/hiera.yaml service_names | grep -q nova_libvirt
              then
                GPFS_VOLUME_PATH=GPFS_VOLUME_SHARE/`echo -n GPFS_NFS_SERVER:GPFS_VOLUME_SHARE | md5sum | awk '{print $1; exit}'` printf "GPFS_NFS_SERVER:GPFS_VOLUME_SHARE $GPFS_VOLUME_PATH nfs4 GPFS_NOVA_NFS_OPTIONS 0 0\n" >> /etc/fstab
                mount -a
              else
                GPFS_VOLUME_PATH=GPFS_VOLUME_SHARE/`echo -n GPFS_NFS_SERVER:GPFS_VOLUME_SHARE | md5sum | awk '{print $1; exit}'` printf "GPFS_NFS_SERVER:GPFS_VOLUME_SHARE $GPFS_VOLUME_PATH nfs4 GPFS_CINDER_NFS_OPTIONS 0 0\n" >> /etc/fstab
                mount -a
              fi
            fi
          params:
            GPFS_VOLUME_SHARE: {get_param: SpectrumScaleVolumeShare}
            GPFS_NFS_SERVER: {get_param: SpectrumScaleNFSServer}
            GPFS_CINDER_NFS_OPTIONS: {get_param: SpectrumScaleCinderNFSOptions}
            GPFS_NOVA_NFS_OPTIONS: {get_param: SpectrumScaleNovaNFSOptions}
  CustomExtraDeployments:
    type: OS::Heat::SoftwareDeploymentGroup
    properties:
      servers: {get_param: servers}
      config: {get_resource: CustomExtraConfig}
      actions: ['CREATE', 'UPDATE']
      input_values:
        deploy_identifier: {get_param: DeployIdentifier}

```

Chapter 18. Configuring Multi-Rail over TCP (MROT)

IBM Storage Scale 5.1.5 introduces the Multi-Rail over TCP (MROT) feature. This feature enables the concurrent use of multiple subnets to communicate with a specified destination, and now allows the concurrent use of multiple physical network interfaces without requiring bonding to be configured.

With MROT, the subnets attribute in the **mmchconfig** command can be used to establish fault tolerance or automatic failover. All the IP addresses which are in the subnets attribute you define are used to establish connections with the nodes within the cluster. If some of the interfaces corresponding to these IP addresses are down, GPFS uses the other subnet-defined interfaces for communication. It is necessary that the interfaces corresponding to the daemon IP addresses are operational even with the subnets attribute configured. This requirement is similar to releases where MROT is not implemented.

In addition to the IP addresses in the subnets attribute, if you also want to use the daemon IP address for communication you need to configure the subnet of daemon IP address in the subnets attribute as well. In releases where MROT is not implemented, only one IP address is used for daemon communication. It is either the daemon IP address or another IP address that is taken from the subnets attribute.

With MROT, when the subnets attribute lists multiple subnets, if any of these multiple subnets are defined on both the local node and the peer node, then all the common subnets are used for communication. In versions where MROT is not implemented, only the first subnet that is common in the list is used for communication between the local and peer nodes.

Rules for configuring the subnets attribute

Use the **mmchconfig** command to modify the subnets attribute, to define the IP addresses and network interfaces used for daemon communication. The following rules define how the subnets attribute configuration is processed for any local node, and how communication is done with the peer nodes:

- If the subnets attribute is not configured, daemon IP addresses are used for communication.
- Multiple subnets can be configured and will be used for communication if the peer node has the network interfaces configured on all the common subnets.
- All IP addresses that are defined in the common subnets are used for communication.
- TCP connections are only established between those IP addresses that are within the same subnet.
- If IP addresses are defined in the subnets attribute for both the local and the peer node in the same subnet, then the daemon IP address is not used for communication, by default. To use the daemon IP address as well, it must be configured through the subnets attribute and the daemon IP addresses of a pair of nodes must be in the same subnet.
- If there are no IP addresses in the same subnet, then daemon IP addresses are used for communication.
- Load balance and failover are supported.

Failure detection and recovery

When a TCP connection is marked as broken, ICMP is used to detect the connectivity of the source IP address and the destination IP address. If the ICMP echo reply is not received, the IP pair is marked as down. A background recovery thread continuously detects the connectivity of this IP pair. When the ICMP echo reply is received, this IP pair is marked as up, and can be used again to establish TCP connections. The ICMP echo command (network ping) must be unblocked in the firewall for IBM Storage Scale to function properly.

For more information, see “[Firewall recommendations for internal communication among nodes](#)” on page 985.

IP pair table

With MROT, the IP pair table contains all the IP pairs, including the source IP address and the destination IP address which are used to establish TCP connections between the local node and the peer node. When the subnets attribute lists multiple subnets, and if any of these multiple subnets are defined for both the local node and the peer node, then all the common subnets are used for communication. All the IP addresses in the local node and the peer node which are defined in the common subnets are used for communication and are listed in the IP pair table. You can use the `mmdiag --network` option to show the IP pair table. For example, specifying the following subnets:

```
subnets='192.168.1.0 10.0.0.0'
```

For the IP pair table between node A and node B, if node A has network interfaces with IP addresses 192.168.1.1 and 10.0.0.1, and node B has network interfaces with IP addresses 192.168.1.2 and 10.0.0.2, then the IP pair table on node A will be as shown.

Table 13. IP pair table on node A					
idx	iface	Status	Source	Destination	Subnet
0	eth0	up	10.0.0.1	10.0.0.2	10.0.0.0/24
1	eth1	up	192.168.1.1	192.168.1.2	192.168.1.0/24

The TCP connections between node A and node B are established based on the IP pair table, by using a round-robin algorithm.

Configuring N:N connection model

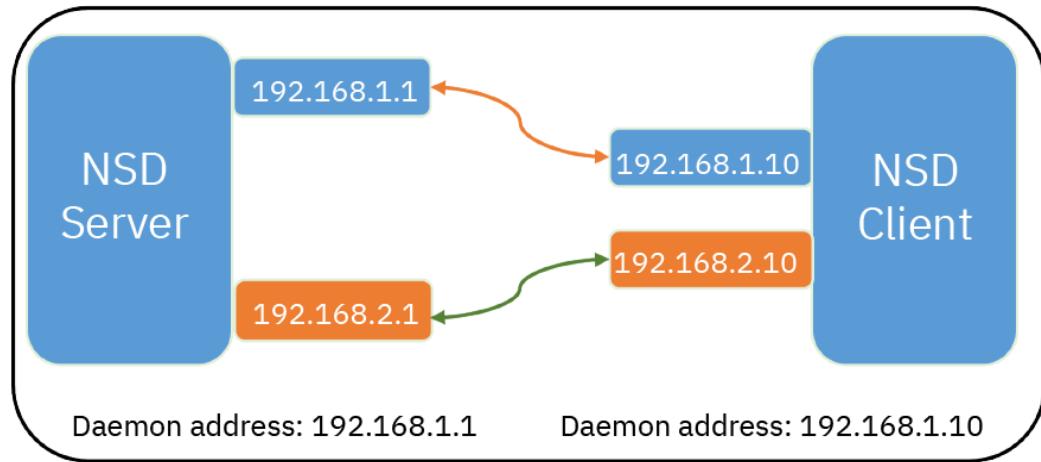


Figure 6. Configuring the N:N connection model

For a specific subnet in the subnets attribute list, when the number of IP addresses in this subnet on node A is the same as the number of IP addresses in this subnet on node B, then that connection model is referred to as the N:N model. There are N IP pairs in this subnet. Figure 1 shows an example of a N:N model on specifying the following subnets:

```
subnets = "192.168.1.0 192.168.2.0"
```

For subnet 192.168.1.0, the NSD server has the network interface with IP address 192.168.1.1, and the NSD client has the network interface with IP address 192.168.1.10. The number of IP addresses on the NSD server is the same as the number of IP addresses on the NSD client. Therefore, this is an N:N connection model. The same applies to subnet 192.168.2.0. The IP pair table on the NSD server is shown in the following table.

Table 14. IP pair table on NSD server for an N:N connection model

idx	iface	Status	Source	Destination	Subnet
0	eth0	up	192.168.1.1	192.168.1.10	192.168.1.0/24
1	eth1	up	192.168.2.1	192.168.2.10	192.168.2.0/24

Configuring M*N connection model

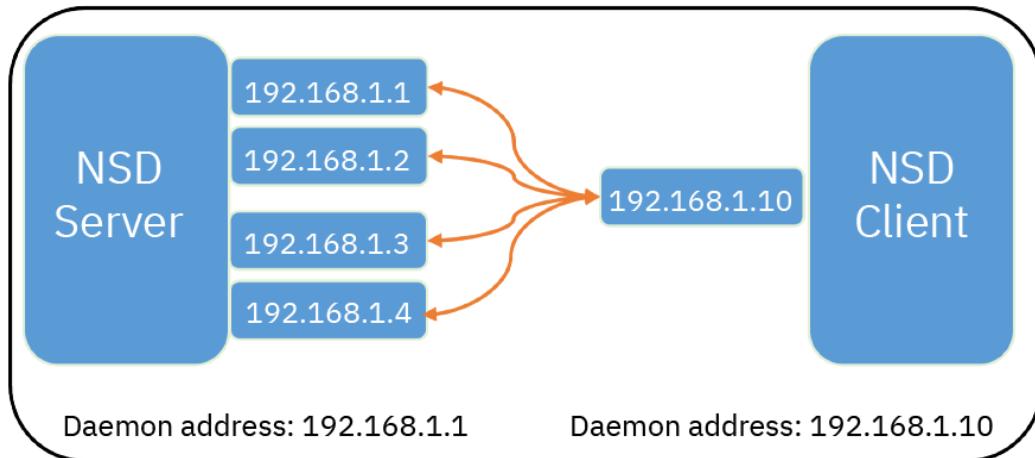


Figure 7. Configuring the M*N connection model

For a specific subnet in the subnets attribute list, when the number of IP addresses in this subnet on node A is not the same as the number of IP addresses in this subnet on node B then such a connection model is referred to as M*N model. There are M*N IP pairs in this subnet. Figure 2 shows an example, when specifying the following subnets:

```
subnets = "192.168.1.0"
```

For subnet 192.168.1.0, the NSD server has the network interface with the IP addresses 192.168.1.1, 192.168.1.2, 192.168.1.3 and 192.168.1.4, and the NSD client has the network interface with the IP address 192.168.1.10. The number of IP addresses on the NSD server is not the same as the number of IP addresses on the NSD client. Therefore, this is referred to as M*N connection model. The IP pair table on the NSD server is shown.

Table 15. IP pair table on NSD server for an M*N connection model

idx	iface	Status	Source	Destination	Subnet
0	eth0	up	192.168.1.1	192.168.1.10	192.168.1.0/24
1	eth1	up	192.168.1.2	192.168.1.10	192.168.1.0/24
2	eth2	up	192.168.1.3	192.168.1.10	192.168.1.0/24
3	eth3	up	192.168.1.4	192.168.1.10	192.168.1.0/24

Configuring multiple IP addresses in the same subnet

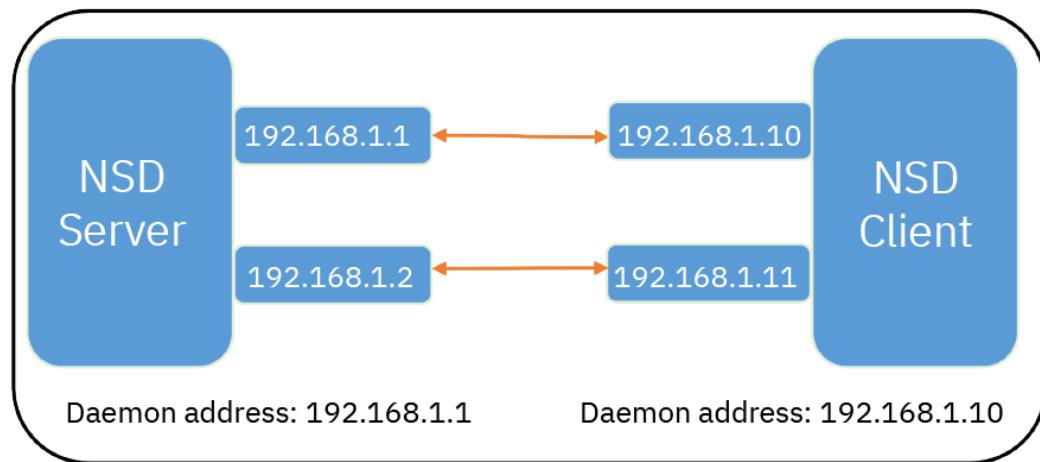


Figure 8. Configuring multiple IP addresses

If multiple IP addresses are configured in the same subnet, then specific OS dependent configurations must be configured. Figure 3 shows an example by specifying the following subnets:

```
subnets = "192.168.1.0"
```

For subnet 192.168.1.0, the NSD server has network interfaces with the IP addresses 192.168.1.1 and 192.168.1.2, and the NSD client has the network interfaces with the IP addresses 192.168.1.10 and 192.168.1.11. The IP pair table on the NSD server is shown:

Table 16. IP pair table on NSD server for multiple IP addresses					
idx	iface	Status	Source	Destination	Subnet
0	eth0	up	192.168.1.1	192.168.1.10	192.168.1.0/24
1	eth1	up	192.168.1.2	192.168.1.11	192.168.1.0/24

The specifications shown in the preceding tables can function properly only if you define certain OS dependent configurations.

Linux configurations

In order to have multiple network interfaces on the same subnet, `arp_filter` and source-based policy routing needs to be configured. The official reference values for configuring the `arp_filter` with the variable as shown:

```
variable: net.ipv4.conf.interface.arp_filter.
```

- 1 - Allows you to have multiple network interfaces on the same subnet, and have the ARPs for each interface be answered based on whether or not the kernel would route a packet from the ARP'd IP out that interface. Therefore, you must use source based routing for this to work. In other words, it allows control over which cards will respond to an ARP request. In most cases, it is 1.
- 0 - This is the default value. The kernel can respond to ARP requests with addresses from other interfaces. This may seem wrong but it usually makes sense, because it increases the chance of successful communication. IP addresses are owned by the complete host on Linux, not by particular interfaces. This behavior might cause problems only for more complex setups like load-balancing.

For more information, see <https://www.kernel.org/doc/Documentation/networking/ip-sysctl.txt>.

The `arp_filter` is enabled for the interface if at least one of the following attributes `conf/` {`all`, `interface`} / `arp_filter` is set to `TRUE`; otherwise it is disabled.

Note: This configuration is defined per interface setting, where “interface” is the name of your network interface; “all” is a special interface. It changes the settings for all interfaces.

Follow the procedure shown to complete the Linux configurations.

On the NSD Server

1. Issue the following command to set the arp_filter.

```
sysctl -w net.ipv4.conf.default.arp_filter=1  
sysctl -w net.ipv4.conf.all.arp_filter=1
```

2. Issue the following commands to set the source-based policy routing.

```
ip addr add 192.168.1.1/24 dev eth0  
ip addr add 192.168.1.2/24 dev eth1  
echo 200 subnet_192.168.1.0_eth0 >> /etc/iproute2/rt_tables  
echo 201 subnet_192.168.1.0_eth1 >> /etc/iproute2/rt_tables  
ip rule add from 192.168.1.1 lookup subnet_192.168.1.0_eth0  
ip rule add from 192.168.1.2 lookup subnet_192.168.1.0_eth1  
ip route add 192.168.1.0/24 dev eth0 table subnet_192.168.1.0_eth0  
ip route add 192.168.1.0/24 dev eth1 table subnet_192.168.1.0_eth1
```

On the NSD Client

1. Issue the following command to set the arp_filter

```
sysctl -w net.ipv4.conf.default.arp_filter=1  
sysctl -w net.ipv4.conf.all.arp_filter=1
```

2. Issue the following command to set the source-based policy routing.

```
ip addr add 192.168.1.10/24 dev eth0  
ip addr add 192.168.1.11/24 dev eth1  
echo 200 subnet_192.168.1.0_eth0 >> /etc/iproute2/rt_tables  
echo 201 subnet_192.168.1.0_eth1 >> /etc/iproute2/rt_tables  
ip rule add from 192.168.1.10 lookup subnet_192.168.1.0_eth0  
ip rule add from 192.168.1.11 lookup subnet_192.168.1.0_eth1  
ip route add 192.168.1.0/24 dev eth0 table subnet_192.168.1.0_eth0  
ip route add 192.168.1.0/24 dev eth1 table subnet_192.168.1.0_eth1
```

Configuring the subnets attribute for multi-cluster

Configuring the subnets attribute for multi-cluster is similar to configuring it for the local cluster. However, in this case, both the home cluster and the remote cluster must be configured. You must also specify a cluster name or a cluster name pattern for each subnet. This is needed when a private network is shared across clusters. If the use of a private network is limited to only the local cluster, then no cluster name is required in the subnets attribute.

Configuring maxTcpConnsPerNodeConn

The total number of TCP connections between the two nodes is controlled by the **maxTcpConnsPerNodeConn** attribute. The valid values are 1-16, with the default being 2. If **maxTcpConnsPerNodeConn** is less than the number of IP pairs between a pair of nodes, only some IP pairs, specifically the ones defined in **maxTcpConnsPerNodeConn**, are used for communication. For example, if **maxTcpConnsPerNodeConn** is 2, but there are 4 IP pairs in the IP pair table between a pair of nodes, then only 2 IP pairs are used for communication.

For more information, see “[Recommendations for tuning maxTcpConnsPerNodeConn parameter](#)” on page [77](#)

Chapter 19. Dynamic pagepool configuration

The dynamic pagepool feature is disabled by default. To enable it, the **dynamicPagepoolEnabled** parameter must be set to yes, and restart IBM Storage Scale on the node.

The following configuration parameters are available to adjust the size of the dynamic pagepool:

pagepoolMinPhysMemPct

Specifies the smallest possible size of the dynamic pagepool as a percentage of the node memory.

pagepoolMaxPhysMemPct

Specifies the largest possible size of the dynamic pagepool as a percentage of the node memory.

pagepoolChangeGracePeriod

Specifies the grace period in seconds between attempts to increase the pagepool.

Note: When the dynamic pagepool feature is enabled on nodes that have more than 16 GiB of memory, the Linux configuration setting, **vm.max_map_count**, must be increased, according to the total memory installed. The appropriate value of **vm.max_map_count** can be determined by dividing the total memory size in bytes by 256 KiB . For example, on a node with 512 GiB of memory, the appropriate value is 2097152, which is the result of 549755813888 divided 262144. The **vm.max_map_count** value can be set by using the Linux **sysctl** command, for example, **sysctl vm.max_map_count=2097152**. To keep this change when the node is rebooted, update the /etc/sysctl.conf file.

- For more information about these parameters, see the *mmchconfig* command in the *IBM Storage Scale: Command and Programming Reference Guide*.
- For more information, see the *Dynamic pagepool* section in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Chapter 20. Configuring shared memory communications direct

Use this procedure to configure the SMC-D on Linux on Z nodes.

1. Verify that the internal shared memory (ISM) device is available. You can list ISM devices by issuing the following command:

```
# lspci | grep ISM
```

A sample output is as follows:

```
1014:00:00.0 Non-VGA unclassified device: IBM Internal Shared Memory (ISM) virtual PCI device
```

For more information about ISM devices, see [Internal shared memory device driver](#).

2. Set up smc-tools and qclib OS packages.

SLES

```
zypper install smc-tools qclib
```

RHEL

```
dnf install smc-tools qclib
```

3. Configure SELinux for SMC-D on RHEL.

SMC sockets are not included in the standard SELinux policies. Therefore, SMC-D does not support the enforcing mode of SELinux with the standard policies. To enable SMC-D support, you must either disable SELinux or install the custom SELinux policy to be able to use the SELinux enforcing mode.

To disable SELinux, complete the following steps.

- a. Verify the SELinux settings in the /etc/selinux/config file.
- b. Set the SELinux=permissive or SELinux=disabled mode.
- c. Restart the node to apply the changes.

To install the custom SELinux policy to be able to use SELinux enforcing mode, complete the following steps.

- a. Install selinux-policy-devel package by issuing the next command:

```
dnf install selinux-policy-devel
```

- b. Create a text file named "storage-scale-smc.te" and add the following content in it:

```
policy_module(storage-scale-smc, 1.0.0)
require {
    type unconfined_service_t;
}
kernel_rw_unlabeled_smcl_socket(unconfined_service_t)
```

- c. Run the following commands to create and instal the policy for SMC sockets:

```
make -f /usr/share/selinux/devel/Makefile storage-scale-smc.pp
semodule -i storage-scale-smc.pp
```

4. Set maximum socket read/write buffer sizes to 1048576 bytes.

```
sysctl -w net.core.rmem_max=1048576 >> /etc/sysctl.conf
```

```
sysctl -w net.core.wmem_max=1048576 >> /etc/sysctl.conf
```

Verifying SMC-D requirements on each node

Check whether all requirements are met before you configure SMC-D on IBM Storage Scale.

1. Verify whether a node meets SMC-D requirements by running SMC-D Prerequisites Verification Tool `tssmcdnodeverify`. It is available on Linux on Z nodes only. The full path is `/usr/lpp/mmfs/bin/tssmcdnodeverify`.

```
# tssmcdnodeverify

Verifying SMC-D requirements on node gpfs01.domain.com
Verifying the platform ...
s390x
Checking HW version ...
Type 3931, IBM z16
Checking OS version ...
RHEL 9.3
Verifying smc-tools package ...
smc-tools-1.8.2-1.el9.s390x
Verifying ISM devices ...
IBM Internal Shared Memory (ISM) virtual PCI device
The node gpfs01.domain.com is ready for SMC-D
```

2. Verify whether a node is running on a specific IBM Z® central processor complex (CPC) by using an optional -c parameter.

```
# tssmcdnodeverify -c M123

Verifying SMC-D requirements on node gpfs01.domain.com
Verifying the platform ...
s390x
Checking HW version ...
Type 3931, IBM z16
Checking OS version ...
RHEL 9.3
Verifying smc-tools package ...
smc-tools-1.8.2-1.el9.s390x
Verifying ISM devices ...
IBM Internal Shared Memory (ISM) virtual PCI device
Verifying qclib package ...
qclib-2.3.2-1.el9.s390x
Checking if the node is running on CPC M123 ...
CPC of the node: M123
The node gpfs01.domain.com is ready for SMC-D
```

3. Verify whether a node can establish an SMC-D connection to other nodes.

```
# mmnetverify smcd
```

A sample output is as follows:

```
gpfs01 checking communication with node gpfs02.
  Operation smcd: Success.
gpfs01 checking communication with node gpfs03.
  Operation smcd: Success.
gpfs01 checking communication with node gpfs01.
  Operation smcd: Success.
No issues found.
```

Chapter 21. Performing GPFS administration tasks

Before you perform GPFS administration tasks, review topics such as getting started with GPFS, requirements for administering a GPFS file system, and common command principles.

For information on getting started with GPFS, see the *IBM Storage Scale: Concepts, Planning, and Installation Guide*. This includes:

- Installing GPFS
- GPFS cluster creation considerations
- Configuring and tuning your system for GPFS
- Starting GPFS
- Network Shared Disk creation considerations
- File system creation considerations

The information for administration and maintenance of GPFS and your file systems is covered in topics that include:

- [“Requirements for administering a GPFS file system” on page 199](#) and [“Common GPFS command principles” on page 201](#)
- [Chapter 1, “Configuring the GPFS cluster,” on page 1](#)
- [Chapter 26, “Managing file systems,” on page 217](#)
- [Chapter 28, “Managing disks,” on page 275](#)
- [Chapter 34, “Managing GPFS quotas,” on page 425](#)
- [Chapter 36, “Managing GPFS access control lists,” on page 461](#)
- [**Command reference** in *IBM Storage Scale: Command and Programming Reference Guide*](#)
- [**GPFS programming interfaces** in *IBM Storage Scale: Command and Programming Reference Guide*](#)
- [**GPFS user exits** in *IBM Storage Scale: Command and Programming Reference Guide*](#)
-
- [Chapter 27, “File system format changes between versions of IBM Storage Scale,” on page 267](#)

Requirements for administering a GPFS file system

Root authority is required to perform all GPFS administration tasks except those with a function that is limited to listing certain GPFS operating characteristics or modifying individual user file attributes.

On Windows, root authority normally means users in the Administrators group. However, for clusters with both Windows and UNIX nodes, only the special Active Directory domain user `root` qualifies as having root authority for the purposes of administering GPFS. For more information about GPFS prerequisites, see the topic *Installing GPFS prerequisites* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

The GPFS commands are designed to maintain the appropriate environment across all nodes in the cluster. To achieve this goal, the GPFS commands use the remote shell and remote file copy commands that you specify on either the `mmcrcluster` or the `mmchcluster` command.

The default remote commands are `ssh` and `scp`, but you can designate any other remote commands provided they have compatible syntax.

In principle, you can issue GPFS administration commands from any node in the cluster. The nodes that you plan to use for administering GPFS must be able to execute remote shell commands on themselves and on any other node in the cluster. They must do so without the use of a password and without producing any extraneous messages. Similarly, the nodes on which the GPFS commands are issued must

be able to copy files to and from any other node in the cluster. And the nodes must do so without the use of a password and without producing any extraneous messages.

The way the passwordless access is achieved depends on the particular remote execution program and authentication mechanism that is used. For example, for `rsh` and `rcp`, you might need a properly configured `.rhosts` file in the root user's home directory on each node in the GPFS cluster. If the remote program is `ssh`, you can use private identity files that do not have a password. Or, if the identity file is password-protected, you can use the `ssh-agent` utility to establish an authorized session before you issue `mm` commands.

You can avoid configuring your GPFS nodes to allow remote access to the root user ID, by using sudo wrapper scripts to run GPFS administrative commands. See “[Running IBM Storage Scale commands without remote root login](#)” on page 26.

GPFS does not need to know which nodes are being used for administration purposes. It is the administrator's responsibility to issue `mm` commands only from nodes that are properly configured and can access the rest of the nodes in the cluster.

Note: If your cluster includes Windows nodes, you must designate `ssh` and `scp` as the remote communication program.

The `adminMode` configuration attribute

GPFS recognizes the `adminMode` configuration attribute. It specifies whether all nodes in the cluster would be used for issuing GPFS administration commands or just a subset of the nodes.

The `adminMode` attribute is set with the `mmchconfig` command and can have one of two values:

allToAll

Indicates that all nodes in the cluster can be used for running GPFS administration commands and that all nodes are able to execute remote commands on any other node in the cluster without the need of a password.

The major advantage of this mode of operation is that GPFS can automatically recover missing or corrupted configuration files in almost all circumstances. The major disadvantage is that all nodes in the cluster must have root level access to all other nodes.

central

Indicates that only a subset of the nodes will be used for running GPFS commands and that only those nodes are able to execute remote commands on the rest of the nodes in the cluster without the need of a password.

The major advantage of this mode of administration is that the number of nodes that must have root level access to the rest of the nodes is limited and can be as low as one. The disadvantage is that GPFS might not be able to automatically recover from loss of certain configuration files. For example, if the SSL key files are not present on some of the nodes, the operator must intervene to recover the missing data. Similarly, it may be necessary to shut down GPFS when adding new quorum nodes. If an operator intervention is needed, you see appropriate messages in the GPFS log or on the screen.

Note List:

1. Any node that is used for the IBM Storage Scale GUI is considered as an administrative node and must have the ability to execute remote commands on all other nodes in the cluster without the need of a password as the root user or as the configured `gpfs admin` user.
2. If the GPFS cluster is configured to support Clustered NFS (CNFS), all CNFS member nodes must belong to the subset of nodes that are able to execute remote commands without the need of a password as the root user or as the configured `gpfs admin` user.
3. If the GPFS cluster is configured to support Clustered export services (CES), all CES member nodes must belong to the subset of nodes that are able to execute remote commands without the need of a password as the root user or as the configured `gpfs admin` user.
4. The IBM Storage Scale REST API must be configured on nodes that are able to execute remote commands without a password as the root user or as the configured `gpfs admin` user.

5. If an IBM Storage Protect server is used to back up the GPFS file system data, the nodes that are used as IBM Storage Protect clients must belong to the subset of nodes that are able to execute remote commands without the need of a password as the root user or as the configured gpfs admin user.
6. Windows GPFS clusters typically use central mode. allToAll mode requires that the GPFS Administrative service (mmwinserv) be configured to run as the special domain root account. For more information, see *Procedure for installing GPFS on Windows nodes* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
7. If Call home is configured to execute daily/weekly data gathering or the autoconfig option is to be used, the call home node (also known as call home server) must be able to reach call home clients without a password for scp data transfer as the root user or as the configured gpfs admin user.

Clusters that are created with the GPFS 3.3 or later level of the code have adminMode set to central by default. Clusters that are migrated from GPFS 3.2 or earlier versions continue to operate as before and would have adminMode set to allToAll.

You can change the mode of operations at any time with the help of the mmchconfig command. For example, to switch the mode of administration from allToAll to central, issue:

```
mmchconfig adminMode=central
```

Use the mmclsconfig adminMode command to display the mode of administration currently in effect for the cluster.

Common GPFS command principles

There are some common principles that you should keep in mind when you are running GPFS commands.

Those principles include:

- Unless otherwise noted, GPFS commands can be run from any node in the cluster. Exceptions are commands that are not supported in a particular operating system environment. Certain commands may additionally require the affected file system to be mounted.
- GPFS supports the "no" prefix on all Boolean type long (or dash-dash) options.

Specifying nodes as input to GPFS commands

Many GPFS commands accept a node or multiple nodes as part of their input by using the -N flag.

Nodes can be specified to GPFS commands in various ways:

Node

A representation of an individual node, which can be any of these:

- Short GPFS administration node interface name.
- Long GPFS administration node interface name.
- Short GPFS daemon node interface name.
- Long GPFS daemon node interface name.
- IP address corresponding to the GPFS daemon node interface.
- GPFS node number.

Node - Node

A node range, indicated by specifying two node numbers separated by a hyphen (-), with the first node number being less than or equal to the second node number. For example, node range 3-8 specifies the nodes with node numbers 3, 4, 5, 6, 7, and 8.

NodeClass

A set of nodes that are grouped into system-defined node classes or user-defined node classes. The system-defined node classes that are known to GPFS are:

all

All of the nodes in the GPFS cluster.

clientNodes

All nodes that do not belong to the managerNodes node class.

localhost

The node on which the command is running.

managerNodes

All nodes in the pool of nodes from which file system managers and token managers are selected.

mount

For commands involving a file system, all of the local nodes on which the file system is mounted (nodes in remote clusters are always excluded, even when they mount the file system in question).

nonquorumNodes

All of the non-quorum nodes in the GPFS cluster.

nsdNodes

All of the NSD server nodes in the GPFS cluster.

quorumNodes

All of the quorum nodes in the GPFS cluster.

User-defined node classes are created with the `mmcrnodeclass` command. After a node class is created, it can be specified as an argument on commands that accept the `-N NodeClass` option. User-defined node classes are managed with the `mmchnodeclass`, `mmdeinodeclass`, and `mmlsnodeclass` commands.

NodeFile

A file that contains a list of nodes. A node file can contain individual nodes or node ranges.

For commands operating on a file system, the stripe group manager node is always implicitly included in the node list. Not every GPFS command supports all of the node specification options described in this topic. To learn what kinds of node specifications are supported by a particular GPFS command, see the relevant command description in *Command reference* in *IBM Storage Scale: Command and Programming Reference Guide*.

Stanza files

The input to a number of GPFS commands can be provided in a file organized in a stanza format.

A stanza is a series of whitespace-separated tokens that can span multiple lines. The beginning of a stanza is indicated by the presence of a stanza identifier as the first token on a line. Stanza identifiers consist of the % (percent sign) character, followed by a keyword, and ending with the : (colon) character. For example, `%nsd:` indicates the beginning of an NSD stanza.

A stanza identifier is followed by one or more stanza clauses describing different properties of the object. A stanza clause is defined as an *Attribute=value* pair.

Lines that start with the # (pound sign) character are considered comment lines and are ignored. Similarly, you can imbed inline comments following a stanza clause; all text after the # character is considered a comment.

The end of a stanza is indicated by one of the following:

- a line that represents the beginning of a new stanza
- a blank line
- a non-comment line that does not contain the = character

GPFS recognizes a number of stanzas:

%nsd:

NSD stanza

%pdisk:
Physical disk stanza

%vdisk:
Virtual disk stanza

%da:
Declustered array stanza

%rg:
Recovery group stanza

The details are documented under the corresponding commands.

For more information about the IBM Storage Scale RAID commands that use stanzas, see *IBM Storage Scale RAID: Administration* in Elastic Storage Server (ESS) documentation.

A stanza file can contain multiple types of stanzas. Commands that accept input in the form of stanza files expect the stanzas to be syntactically correct but will ignore stanzas that are not applicable to the particular command. Similarly, if a particular stanza clause has no meaning for a given command, it is ignored.

For backward compatibility, a stanza file may also contain traditional NSD descriptors, although their use is discouraged.

Here is what a stanza file may look like:

```
# Sample file containing two NSD stanzas

# Example for an NSD stanza with imbedded comments
%nsd:    nsd=DATA5      # my name for this NSD
          device=/dev/hdisk5  # device name on node k145n05
          usage=dataOnly
          # List of server nodes for this disk
          servers=k145n05,k145n06
          failureGroup=2
          pool=dataPoolA

# Example for a directly attached disk; most values are allowed to default
%nsd:    nsd=DATA6      device=/dev/hdisk6  failureGroup=3
```

Note: The server name used in the NSD stanza file must be resolvable by the system.

Listing active IBM Storage Scale commands

You can list the active IBM Storage Scale commands that are running on the file system manager node.

Most IBM Storage Scale commands run within the GPFS daemon on the file system manager node. Even if you start a command on another node of the cluster, the node typically sends the command to the file system manager node to be executed. (Two exceptions are the `mmdiag` command and the `mmfsadm` dump command, which run on the node where they were started.)

To list the active commands on the file system manager node, follow these steps:

1. Enter the `mmlsmgr` command with no parameters to discover which node is the file system manager node.

For more information on other options available for the `mmlsmgr` command, see *mmlsmgr command* in *IBM Storage Scale: Command and Programming Reference Guide*.

In the following example, the `mmlsmgr` command reports that node05 is the file system manager node:

```
# mmlsmgr
file system      manager node
-----
gpfs1           192.168.145.14 (node05)

Cluster manager node: 192.168.145.13 (node03)
```

2. Go to the command console on the file system manager node and enter `mmdiag --commands`:

```
# mmdiag --commands
== mmdiag: commands ==
CrHashTable 0x1167A28F0 n 2
  cmd sock 24 cookie 2233688162 owner 38076509 id 0x3FE6046C2700000D(#13) uses 1
  type 76 start 1460415325.957724 flags 0x106 SG none line 'mmdiag --commands'
    cmd sock 12 cookie 521581069 owner 57606185 id 0x3FE6046C2700000C(#12) uses 1
    type 13 start 1460415323.336314 flags 0x117 SG gpfs1 line 'mmrestripefs /dev/business1 -m'
```

The output indicates that two commands are running: the `mmdiag --commands` command that you just entered and the `mmrestripefs` command, which was started from another node.

Note: The output contains two lines about active commands. Each line begins with the term `cmd` and wraps to the next line. You might be interested in the following fields:

start

The system time at which the command was received.

SG

The name of the file system, or None.

line

The command as received by the GPFS daemon.

The remaining input is detailed debugging data that is used for product support. For more information on `mmdiag` command output, see the topic *mmdiag command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Determining how long `mmrestripefs` takes to complete

Several factors determine how long the `mmrestripefs` command takes to complete.

To determine how long the `mmrestripefs` command takes to complete, consider these points:

1. The amount of data that potentially needs to be moved. You can estimate this value by issuing the `df` command.
2. The number of IBM Storage Scale client nodes that are available to do the work.
3. The amount of Network Shared Disk (NSD) server bandwidth that is available for I/O operations.
4. The quality of service for I/O operations (QoS) settings on each node. For more information, see `mmchqos` in the *IBM Storage Scale: Command and Programming Reference Guide*.
5. The maximum number of PIT threads on each node. For more information, see the description of the `pitWorkerThreadsPerNode` attribute in the topic *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide*.
6. The amount of free space that is available from new disks. If you added new disks, issue the `mmdf` command to determine the amount of additional free space that is available.

The restriping of a file system is done by having multiple threads on each node in the cluster work on a subset of files. If the files are large, multiple nodes can participate in restriping it in parallel. So, the more GPFS client nodes that are performing work for the restripe operation, the faster the `mmrestripefs` command completes. Use the `-N` parameter to specify the nodes to participate in the restripe operation. Based on raw I/O rates, you can estimate the length of time for the restripe operation. However, because of the need to scan metadata, double that value.

Assuming that enough nodes are available to saturate the disk servers and assuming that all the data must be moved, the time to read and write every block of data is roughly:

```
2 * fileSystemSize / averageDiskserverDataRate
```

As an upper bound, because of the need to scan all of the metadata, double this time. If other jobs are loading the NSD servers heavily, this time might increase even more.

Note: You do not need to stop all other jobs while the `mmrestripefs` command is running. The CPU load of the command is minimal on each node and only the files that are being restriped at any moment are locked to maintain data integrity.

Chapter 22. Performing parallel copy with mmxcp command

Use the **mmxcp enable** command to perform parallel copies of files from a source directory to a target directory in a single IBM Storage Scale cluster. The copy can occur within a single file system or across different file systems in the same cluster. It can copy from a live file system or from a global or independent fileset snapshot. The **mmxcp** command has a strong relationship with the **mmapplypolicy** command.

The **mmxcp sync** command performs a synchronize operation from the source directory to the target directory. It uses only a single process, but will only try to copy files that are missing or appear to be different.

The **mmxcp verify** command performs a quick compare of the data in the source and target directories. Any difference in the metadata is flagged.

The command also lists configuration information about any currently running **mmxcp** commands and allows you to configure or display the maximum number of **mmxcp** commands that can run at a single time.

For more information on running **mmxcp** command, see the *mmxcp command in the IBM Storage Scale: Command and Programming Reference Guide*.

Storage pools

When you perform parallel copying of a file, the status of storage pool of the file can be one of the following conditions:

- A copied file will be placed into the storage pool that matches the policy rules at the time the file is copied.
- Different file system without storage pool defined, creates copied files in that file system default storage pool.
- Different file system with storage pool defined, creates copied files in that file system default storage pool.

Hardlinks

- The **mmxcp** command supports copying hardlinks (multiple files pointing to the same inode) properly most of the time. However, by default it is limited by the way the policy engine splits up the work for faster parallel execution. By default, the policy engine splits up the work into blocks of 100 entries (files/directories). Any hardlinked files within a block of entries are handled correctly. But, any related hardlinked files within other blocks are handled independently. All files are copied, but not all of them might be pointing to the same inode.
- The **--hardlinks** option can be used to ensure that all hardlinks are copied correctly. It implements a second pass through the source files that targets only hardlinked files, and processes all of them as a single block. The **--hardlinks** option causes the **mmxcp** command to run for a longer period of time.

Fileset

- If the source file system has an independent or dependent fileset and target does not have a fileset, a **subdir** is created with the same name and the files are copied in that **subdir**.

File heat

- If file heat is enabled on the cluster, source files might not have **gpfs.fileHeat EA** but copied files might have due to the copy process generating IO activity on the file.

- Snapshots do not store `gpfs.fileHeat` EAs.

The DMAPI extended attributes are not copied because the file loses the migrated state if copied by using the `--copy-migrated` flag.

File clone

- File clones and their relationships are not preserved by the **mmxcp** command. All files are copied but the copied clone files will now consume additional disk space.
- The `gpfs.CLONE` EA is not copied.

File compression

- The sync does not enforce the file compression. The target files are not compressed.
- By default, the enable target files are not compressed.
- If any of the source files are compressed using **mmchattr**, then you should specify the `--copy-attrs` compression option to compress the target files after the **cp**. The target file compression will use the same compression library that was used for the source.
- If the source file is marked as `illcompressed` and the compression attribute is specified, then the target file will be compressed.
- This may cause the **mmxcp** command to execute for a longer period of time.

File appendonly and immutable attributes

- The **mmxcp sync** command does not copy the appendonly and immutable attributes.
- By default the **mmxcp enable** command does not copy the appendonly and immutable attributes.
- The appendonly and/or immutable attributes can be copied by using the `--copy-attrs` option and specifying appendonly and/or immutable when running the **mmxcp enable** command.
- The `--copy-attrs` option might take longer time to run the **mmxcp** command.
- If there are preexisting files in the target directory that have the appendonly or immutable attributes set, an error might occur. Because these files cannot be overwritten until the appendonly or immutable attributes are removed. See the **mmchattr** command in the *IBM Storage Scale: Command and Programming Reference Guide*.

Policy engine interactions

The **mmxcp** command calls the **mmapapplypolicy** command directly and uses the policy engine LIST rule functions to execute the `/usr/lpp/mmfs/bin/xcputil.sh` script.

See *mmapapplypolicy command in IBM Storage Scale: Command and Programming Reference Guide* for information about functional and performance hints of the following **mmxcp** flags. These flags are passed directly to the **mmapapplypolicy** command.

You can use `-g` option to specify a global work directory or `-s` option to specify a local work directory in which one or more nodes can store temporary files. Otherwise, you can use `-N` option to specify a set of nodes to run parallel instances of policy code for better performance. To display the number of threads that are created and dispatched within each **mmapapplypolicy** process, you can use `-m` option. The `-a` option specifies the number of threads and sort pipelines each node will run during the parallel inode scan and policy evaluation or you can use `-n` option to display number of threads that will be created and dispatched within each **mmapapplypolicy** process during the directory scan phase.

To control the number of files that are passed for each invocation of the EXEC script, you can use `-B` option and to control the level of information displayed by the **mmapapplypolicy** command, you can use `-L n` option. The `--sort-buffer-size` option can be used to set the sort-buffer size that is passed to the **sort** command. The `--qos` option specifies the Quality of Service for I/O operations (QoS) class to which the instance of the command is assigned.

General specifications

- Both the source file system and the destination file system must be mounted on the node where the command is initiated.
- The same enable cannot be run at the same time. It is verified that the source and destination do not already exist in a copy operation that is running in the cluster.
- The same sync cannot be run at the same time. It is verified that the source and destination do not already exist in a sync operation that is running in the cluster.
- By default, a maximum 10 concurrent copy or sync operations can be running on the cluster at any one time. This variable can be set by using the **mmxcp config** command.
- By default, any other Linux nodes in the cluster with the source file system and destination file system both mounted might also be used for the copy command. If the -N option is included, then the subset of nodes with both file systems mounted included through the -N option will be used.
- Error logging from the command will be written to the log file: /var/adm/ras/mmxcp.log. Errors from other nodes in the command will be written in the mmxcp.log on the node where the code is run.
- Messages from the policy engine going to standard out are redirected to the log file: /var/adm/ras/mmxcp.log. Messages to standard error are shown to the screen.
- Currently, no message is given that files were skipped due to being in migrated state.
- Need to execute --copy-migrated with -N where IBM Storage Protect client is installed.
- When copying files from a live file system be aware that files that are created during the **mmxcp** execution might not be copied and files that are deleted during the **mmxcp** execution might cause the **mmxcp** command fail.
- Use the **mmxcp verify** option to perform a quick compare of metadata between a source/snapshot and a target directory from a previously executed **mmxcp** copy or sync.
- When migrated files are synced, it will recall the files before copying the files.
- Examine the /var/adm/ras/mmxcp.log file to see the total number of files updated during the sync command.

Chapter 23. Managing shared memory communications direct

You can enable or disable and set buffer size on an IBM Storage Scale cluster.

Enabling SMC-D on a cluster

By default SMC-D is disabled on IBM Storage Scale. To enable SMC-D on a cluster, complete the following steps:

1. Set the **tscSmcD** parameter to yes.

```
# mmchconfig tscSmcD=yes
```

2. Stop the GPFS daemons.

```
# mmshutdown -a
```

3. Start the GPFS daemons.

```
# mmstartup -a
```

4. Check the SMC-D status.

```
# mmlsconfig tscSmcD
```

The sample output is as follows:

```
tscSmcD yes
```

SMC-D is enabled.

Disabling SMC-D on a cluster

To disable SMC-D on a cluster, complete the following steps:

1. Set the **tscSmcD** parameter to no.

```
# mmchconfig tscSmcD=no
```

2. Stop the GPFS daemons.

```
# mmshutdown -a
```

3. Start the GPFS daemons.

```
# mmstartup -a
```

4. Check the SMC-D status.

```
# mmlsconfig tscSmcD
```

The sample output is as follows:

```
tscSmcD no
```

SMC-D is disabled.

Tuning SMC-D buffer sizes

In IBM Storage Scale, 512K is the default value for SMC-D Send and Receive buffer sizes. The default value of 512K for SMC-D Send and Receive buffer size is achieved by setting the IBM Storage Scale **socketRcvBufferSize** or **socketSndBufferSize** variables to **0** (or by not setting it). This value is sufficient for most scenarios. If required, you can change SMC sockets buffer sizes by using the commands:

```
mmchconfig socketRcvBufferSize=<receive buffer size in bytes>
mmchconfig socketSndBufferSize=<send buffer size in bytes>
```

Chapter 24. Protecting file data: IBM Storage Scale safeguarded copy

IBM Storage Scale 5.1.5 introduces the safeguarded copy (SGC) as a mechanism to protect file system data.

The safeguarded copy (SGC) is a mechanism to protect data in IBM Storage Scale file systems. It is based on the immutable snapshot feature. The mechanism secures data from deliberate or accidental compromise.

The following facts help to explicitly define the Safeguarded copies:

- Safeguarded copies (SGCs) are immutable copies of a file system or fileset, and they help minimize the impact of cyberattacks, disasters and failures.
- SGCs are snapshots with a retention time, and which cannot be altered, and also cannot be deleted until the retention time is expired.
- The administrator can use the GUI to schedule periodic snapshot creation, and also snapshot deletion after the retention time has expired.
- An SGC cannot be modified or deleted by a single bad actor, and the SGC's retention time cannot be changed by a single bad actor.

Safeguarded copies are used to take frequent snapshots of a production file system. You can take, for example, hourly snapshots that are maintained for a number of days. These snapshots act as backup copies for recovering data if the primary data is corrupted or destroyed. If the content of files is damaged, you can run the **mmrestorefs** command to restore the production file system or its independent filesets from these immutable snapshots. These immutable snapshots share the storage space with the production file system, and can be accessed online along with other regular snapshots. The system administrator can also make safeguarded copies offline by copying data out to a separate storage system from these immutable snapshots, then restore the production file system from there, on demand.

As a safeguarded copy, an immutable snapshot cannot be modified or deleted. You can set the retention period for a snapshot when the snapshot is created, which prevents the snapshot from being deleted until the retention time is completed or expires. This feature helps protect the data even if the administrator's account is compromised. By periodically creating snapshots with a retention period, one always ensures ready availability of a snapshot to restore the content of the data, if needed.

For security reasons, the safeguarded copy environment must be managed by a non-root administrator to prevent a malicious attacker from acquiring root user privilege by taking over the administrator account, and corrupting the production file system or disks with operations allowed only for root user. Several IBM Storage Scale operations require root privilege. Therefore, for a safeguarded copy environment, you must configure the IBM Storage Scale sudo wrapper. For more information, see <https://community.ibm.com/community/user/storage/blogs/nils-haustein1/2020/12/17/spectrum-scale-sudo-wrappers>.

Configuring an IBM Storage Scale administrator to run using a non-root account has the benefit of not allowing the administrator to alter the system clock. If allowed to manipulate the system clock, administrators can move the time of day forward past the snapshot's expiration time, and delete the snapshot, therefore defeating the use of snapshot retention time.

In certain cases, such as when the file system runs out of space, it might become necessary to delete a safeguarded copy snapshot even before its retention time expires. In those cases, the **mmrestrictedctl** command can be used to delete snapshots whose retention time has not expired.

To prevent the IBM Storage Scale administrator from removing snapshots until their retention time has expired, the **mmrestrictedctl** command must be run only with explicit authorization from the security administrator. The sudo rules must also be accordingly constructed. The sudo configuration that is mentioned in the blog <https://community.ibm.com/community/user/storage/blogs/nils-haustein1/2020/12/17/spectrum-scale-sudo-wrappers> instructs the **mmrestrictedctl** command to remain disabled for normal administrators. It becomes available only after the security administrator

temporarily adds the command to the allowed list of commands for the IBM Storage Scale administrator. For more information, see *mmrestrictedctl command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Data protection using safeguarded copies

Data protection can be ensured by using the following three workflows:

Establish a safeguarded environment.

Follow the steps to establish a safeguarded copy environment:

1. Prepare the sudo administration environment. For more information, see <https://community.ibm.com/community/user/storage/blogs/nils-haustein1/2020/12/17/spectrum-scale-sudo-wrappers>.
2. In the IBM Storage Scale GUI, schedule periodic snapshot creation with retention time, and deletion of those snapshots after the expiration time has elapsed. For more information, see “[Creating immutable snapshots using the GUI](#)” on page 212.

Use **mmrestorefs** command to restore the content of damaged files

Follow the steps to restore damaged files

1. Stop the applications that are using the corresponding files, filesets or file systems.
2. Use the **mmrestorefs** command to restore data from a snapshot that contains good data into the file system or fileset.

Delete snapshots before their expiration time owing to an emergency

Follow the steps to delete snapshots before their expiration time, in case it becomes necessary.

1. The IBM Storage Scale administrator contacts the security administrator and requests to be granted permission to run the **mmrestrictedctl** command.
2. When permission is granted, the IBM Storage Scale administrator runs the **mmrestrictedctl** command to delete snapshots which have still not expired. The number of snapshots being deleted must be only as few as required.
3. The IBM Storage Scale administrator contacts the security administrator to remove the permission to run the **mmrestrictedctl** command.

Note: The operations that are enabled by the **mmrestrictedctl** command can only be used by running this command and must be used only for certain unusual cases like the ones mentioned. The IBM Storage Scale management GUI does not provide any features to run this command.

Creating immutable snapshots using the GUI

To streamline the management of safeguarded copies, you can configure a rule to automatically create and delete immutable snapshots with the desired frequency and retention period through the IBM Storage Scale management GUI.

To schedule snapshot creation and retention, perform the following steps:

1. Log in to the IBM Storage Scale GUI and select **Files > Snapshots**.
2. Click **Create Snapshot**.
3. In the **Create Snapshot** window, type the path of the file system or independent fileset for which you need to create snapshots.
4. In the **Snapshot name** field, type the name of the snapshot.
5. Click **Create Rule** to schedule the snapshot creation and retention. The **Create Snapshot Rule** window is displayed.
6. In the **Name** field, type the name of the snapshot scheduling rule.
7. In the **Frequency** field, select the frequency in which you need to create snapshot. You need to enter some more details based on the value that is selected in the field. For example, if the **Multiple Times an Hour** value is selected, then select the minutes of the hour in which you need to create snapshots.

8. In the **Retention** fields, select the number of snapshots that must be retained in a period.
9. In the **Deletion Schedule** field, select whether you want the snapshots to be deleted immediately after creation or during off-hours
10. In the **Prefix** field, specify a prefix to be added with the name of the snapshots that are created with this rule. The prefix is added to the date and time to identify the rule that is used to create the snapshot. If a prefix is not specified, the default prefix "@GMT" is used. Using the default prefix enables the Microsoft Windows Volume Shadow Copy Service (VSS) identification if the file is shared using the SMB protocol.
11. Select the **Allow Expiration** checkbox to delete the snapshot when the defined retention period is completed.

Note: If you do not select the **Allow Expiration** checkbox then the expiration time for all snapshots created is the same as its creation time.
12. Click **OK** to save the changes
13. In the **Create Snapshot** window, click **Create** to create the snapshot creation schedule and retention policy for the snapshots.

Restoring data with `mmrestorefs` command

If fileset or file system data is damaged or corrupted, it can be restored from one of the snapshots with the **mmrestorefs** command. If only a few files have been affected then these can be restored by copying them directly from the snapshot to the corresponding file in the file system. The snapshot appears as part of the file system or fileset namespace. For more information, see *mmrestorefs command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Chapter 25. Verifying network operation with the `mmnetverify` command

Verify network operation with the `mmnetverify` command.

Important: Proper operation of IBM Storage Scale depends on reliable TCP/IP communication among the nodes of a cluster. Before you create or reconfigure an IBM Storage Scale cluster, ensure that a proper hostname resolution and ICMP echo (network ping) are enabled among the nodes.

With the `mmnetverify` command, you can do many types of network checks either before or after you create or reconfigure a cluster. Run the command beforehand to verify that the nodes can communicate properly. Run the command afterward at any time to verify communication or to analyze a network problem. For more information, see the topic *mmnetverify command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

The `mmnetverify` command uses the concepts of local nodes and target nodes. A *local node* is a node from which a network test is run. You can enter the command on one node, and have it run on multiple separate local nodes. A *target node* is a node against which a test is run.

You can run tests on one node against multiple nodes. The following command runs tests on node1 against node2 and then on node1 against node3:

```
mmnetverify connectivity -N node1 --target-nodes node2,node3
```

You can also run tests on multiple nodes against multiple nodes. The following command runs tests on node1 against node1 and node2 and then on node2 against node1 and node2:

```
mmnetverify connectivity -N node1,node2 --target-nodes node1,node2
```

It is not necessary to enter the command from a node that is involved in testing. For example, you can run the following command from node1, node2, node3, or any other node in the cluster:

```
mmnetverify data -N node1 --target-nodes node2,node3
```

To run tests against all the nodes in the cluster, omit the `--target-nodes` parameter (example 1). Similarly, to run the test on all the nodes in the cluster, omit the `-N` parameter (example 2):

```
(1) mmnetverify data-medium -N node1  
(2) mmnetverify data-medium --target-nodes node2,node3,node4
```

To run all the tests, omit the test parameter:

```
mmnetverify -N node1 --target-nodes node2,node3,node4
```

The groups of tests include connectivity, port, data, bandwidth, and flood tests. You can run tests individually or as a group. For example, you can run resolution, ping, shell, and copy tests individually, or you can run all of them by specifying the keyword `connectivity`.

The command writes the results of tests to the console by default, or to a log file as in the following example:

```
mmnetverify port -N node1 --target-nodes all --log-file results.log
```

If you are running tests against nodes that are not organized into a cluster, you must specify the nodes in a configuration file. The file must at minimum contain a list of the nodes in the test. You must also include the node from which you are starting the command:

```
node node_starting  
node node1  
node node2
```

```
node node3  
node node4
```

Run the command in the usual way and include the configuration file:

```
mmnetverify ping -N node1,node2,node3,node4 --target-nodes  
node1,node2,node3,node4 --configuration-file config.txt
```

You can also use the configuration file for other purposes, such as specifying a nondefault shell command or file copy command.

Related information

See the topic [*mmnetverify command*](#) in the [*IBM Storage Scale: Command and Programming Reference Guide*](#).

Chapter 26. Managing file systems

There are several file system management tasks outlined in this topic.

For information on how to create GPFS file systems, see *A sample file system creation in IBM Storage Scale: Concepts, Planning, and Installation Guide* and the *mmcrfs* command in *IBM Storage Scale: Command and Programming Reference Guide*.

File system management tasks include:

1. [“Mounting a file system” on page 217](#)
2. [“Unmounting a file system” on page 221](#)
3. [“Deleting a file system” on page 222](#)
4. [“Determining which nodes have a file system mounted” on page 223](#)
5. [“Checking and repairing a file system” on page 223](#)
6. [“Listing file system attributes” on page 228](#)
7. [“Modifying file system attributes” on page 229](#)
8. [“Querying and changing file replication attributes” on page 229](#)
9. [“Using Direct I/O on a file in a GPFS file system” on page 231](#)
10. [“Restriping a GPFS file system” on page 240](#)
11. [“Querying file system space” on page 241](#)
12. [“Querying and reducing file system fragmentation” on page 242](#)
13. [“Protecting data in a file system using backup” on page 244](#)
14. [“Scale Out Backup and Restore \(SOBAR\)” on page 251](#)

Managing filesets, storage pools and policies is also a file system management task. For more information on managing storage pools, filesets and policies, see Chapter 39, “[Information lifecycle management for IBM Storage Scale](#),” on page 525. Use the following information to manage file systems in IBM Storage Scale.

Managing file system through GPFS GUI

To work with this function in the GUI, log on to the IBM Storage Scale GUI and select **Files > File Systems**. For more information on managing file systems through GUI, see [“Creating and managing file systems by using GUI” on page 261](#).

Mounting a file system

You must explicitly mount a GPFS file system if this is the first time the file system is being mounted after its creation, or you specified *not to* automatically mount (-A no) the file system when you created it.

If you allowed the default value for the automatic mount option (-A yes) when you created the file system, then you do not need to use this procedure after restarting GPFS on the nodes.

To mount a GPFS file system, enter:

```
mmmount device
```

where *device* is the name of the file system. For example, to mount the file system **fs1**, enter:

```
mmmount fs1
```

Mounting a file system on multiple nodes

This topic describes how to mount a file system on multiple nodes.

To mount file system **fs1** on all nodes in the GPFS cluster, issue this command:

```
mmmount fs1 -a
```

To mount a file system only on a specific set of nodes, use the **-N** flag of the **mmmount** command.

Related tasks

[Mounting a file system through GUI](#)

You can use the IBM Storage Scale GUI to mount or unmount individual file systems or multiple file systems on the selected nodes. Use the **Files > File Systems**, **Files > File Systems > View Details > Nodes**, or **Nodes > View Details > File Systems** page in the GUI to mount or unmount a file system.

[Changing a file system mount point on protocol nodes](#)

If required, you can change a file system mount point on IBM Storage Scale protocol nodes.

Related reference

[Mount options specific to IBM Storage Scale](#)

Mount options specific to IBM Storage Scale can be specified with the **-o** parameter on the **mmchfs**, **mmremoteefs**, **mmmount** and **mount** commands. Options specified with the **mmchfs** and **mmremoteefs** commands are recorded in the GPFS configuration files and are passed as default options to subsequent mount commands on all nodes in the cluster. Options specified with the **mmmount** or **mount** commands override the existing default settings and are not persistent.

Mount options specific to IBM Storage Scale

Mount options specific to IBM Storage Scale can be specified with the **-o** parameter on the **mmchfs**, **mmremoteefs**, **mmmount** and **mount** commands. Options specified with the **mmchfs** and **mmremoteefs** commands are recorded in the GPFS configuration files and are passed as default options to subsequent mount commands on all nodes in the cluster. Options specified with the **mmmount** or **mount** commands override the existing default settings and are not persistent.

All of the mount options can be specified using the **-o** parameter. Multiple options should be separated only by a comma. If an option is specified multiple times, the last instance is the one that takes effect. Certain options can also be set with specifically designated command flags. Unless otherwise stated, mount options can be specified as:

option or *option=1* or *option=yes* - to enable the option

nooption or *option=0* or *option=no* - to disable the option

The *option={1 | 0 | yes | no}* syntax should be used for options that can be intercepted by the **mount** command and not passed through to GPFS. An example is the **atime** option in the Linux environment.

The GPFS-specific mount options are:

atime

Update inode access time for each access. This option can also be controlled with the **-S** option of the **mmcrlfs** and **mmchfs** commands.

mtime

Always return accurate file modification times. This is the default. This option can also be controlled with the **-E** option on the **mmcrlfs** and **mmchfs** commands.

noatime

Do not update inode access times on this file system. This option can also be controlled with the **-S** option on the **mmcrlfs** and **mmchfs** commands.

nomtime

Update file modification times only periodically. This option can also be controlled with the **-E** option on the **mmcrlfs** and **mmchfs** commands.

norelatime

Update inode access time for each access. This option is the default if **minReleaseLevel** is less than 5.0.0 when the file system is created. This option can also be controlled with the -S option of the `mmcrfs` and `mmchfs` commands. For more information, see the topic *atime values* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

nosyncnfs

Do not commit metadata changes coming from the NFS daemon synchronously. Normal file system synchronization semantics apply. On AIX nodes, `nosyncnfs` is the default. On Linux nodes, `syncnfs` is the default.

relatime

Allow the update of inode access time only if one of the following is true:

- The existing access time is older than 24 hours. Access time is user configurable through the `atimeDeferredSeconds` configuration attribute.
- The existing file modification time is greater than the existing access time.

This option is the default if **minReleaseLevel** is 5.0.0 or greater when the file system is created.

This option can also be controlled with the -S option of the `mmcrfs` and `mmchfs` commands. For more information, see the topic *atime values* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

syncnfs

Synchronously commit metadata changes coming from the NFS daemon. On Linux nodes, `syncnfs` is the default. On AIX nodes, `nosyncnfs` is the default.

useNSDserver={always | asfound | asneeded | never}

Controls the initial disk discovery and failover semantics for NSD disks. The possible values are:

always

Always access the disk using the NSD server. Local dynamic disk discovery is disabled.

asfound

Access the disk as found (the first time the disk was accessed). No change of disk access from local to NSD server, or the other way around, is performed by GPFS.

asneeded

Access the disk any way possible. This is the default.

never

Always use local disk access.

Related tasksMounting a file system on multiple nodes

This topic describes how to mount a file system on multiple nodes.

Mounting a file system through GUI

You can use the IBM Storage Scale GUI to mount or unmount individual file systems or multiple file systems on the selected nodes. Use the **Files > File Systems**, **Files > File Systems > View Details > Nodes**, or **Nodes > View Details > File Systems** page in the GUI to mount or unmount a file system.

Changing a file system mount point on protocol nodes

If required, you can change a file system mount point on IBM Storage Scale protocol nodes.

Mounting a file system through GUI

You can use the IBM Storage Scale GUI to mount or unmount individual file systems or multiple file systems on the selected nodes. Use the **Files > File Systems**, **Files > File Systems > View Details > Nodes**, or **Nodes > View Details > File Systems** page in the GUI to mount or unmount a file system.

The GUI has the following options related to mounting the file system:

1. Mount local file systems on nodes of the local IBM Storage Scale cluster.
2. Mount remote file systems on local nodes.

3. Select individual nodes, protocol nodes, or nodes by node class while selecting nodes on which the file system needs to be mounted.
4. Prevent or allow file systems from mounting on individual nodes.

Do the following to prevent file systems from mounting on a node:

- a. Go to **Nodes**.
 - b. Select the node on which you need to prevent or allow file system mounts.
 - c. Select **Prevent Mounts** from the **Actions** menu.
 - d. Select the required option and click **Prevent Mount** or **Allow Mount** based on the selection.
5. Configure automatic mount option. The automatic configure option determines whether to automatically mount file system on nodes when GPFS daemon starts or when the file system is accessed for the first time. You can also specify whether to exclude individual nodes while enabling the automatic mount option. To enable automatic mount, do the following:
 - a. From the **Files > File Systems** page, select the file system for which you need to enable automatic mount.
 - b. Select **Configure Automatic Mount** option from the **Actions** menu.
 - c. Select the required option from the list of automatic mount modes.
 - d. Click **Configure**.

Note: You can configure automatic mount option for a file system only if the file system is unmounted from all nodes. That is, you need to stop I/O on this file system to configure this option. However, you can include or exclude the individual nodes for automatic mount without unmounting the file system from all nodes.

Related tasks

[Mounting a file system on multiple nodes](#)

This topic describes how to mount a file system on multiple nodes.

[Changing a file system mount point on protocol nodes](#)

If required, you can change a file system mount point on IBM Storage Scale protocol nodes.

Related reference

[Mount options specific to IBM Storage Scale](#)

Mount options specific to IBM Storage Scale can be specified with the -o parameter on the mmchfs, mmremotefs, mmmount and mount commands. Options specified with the mmchfs and mmremotefs commands are recorded in the GPFS configuration files and are passed as default options to subsequent mount commands on all nodes in the cluster. Options specified with the mmmount or mount commands override the existing default settings and are not persistent.

Changing a file system mount point on protocol nodes

If required, you can change a file system mount point on IBM Storage Scale protocol nodes.

To change a file system mount point on protocol nodes, perform the following steps:

1. Unmount the file system:

```
mmumount fs0 -a
```

2. Change the mount point:

```
mmchfs fs0 -T /ibm/new_fs0
```

3. Change the path of all NFS and SMB exports.

Note: The **mnnfs export change** and the **mssmb export change** commands do not allow path names to be edited. Therefore, the export needs to be removed and re-added.

4. Change the object CCR files:

- account-server.conf
- container-server.conf
- object-server-.conf
- object-server-sof.conf
- spectrum-scale-object.comf
- spectrum-scale-objectizer.conf

The parameter that you need to change varies depending on the configuration file.

- a. Use the **mmobj config change** command to list the parameters for the file. For example, to list the parameters for the object-server.conf file, enter:

```
mmobj config list --ccrfile object-server.conf --section DEFAULT --property devices
```

- b. Use the **mmobj config change --ccrfile** *file name* to change the parameter. For example, to change the object-server.conf file, enter:

```
mmobj config change --ccrfile object-server.conf --section DEFAULT --property devices /newFS/name
```

Related tasks

[Mounting a file system on multiple nodes](#)

This topic describes how to mount a file system on multiple nodes.

[Mounting a file system through GUI](#)

You can use the IBM Storage Scale GUI to mount or unmount individual file systems or multiple file systems on the selected nodes. Use the **Files > File Systems**, **Files > File Systems > View Details > Nodes**, or **Nodes > View Details > File Systems** page in the GUI to mount or unmount a file system.

Related reference

[Mount options specific to IBM Storage Scale](#)

Mount options specific to IBM Storage Scale can be specified with the -o parameter on the mmchfs, mmremoteefs, mmmount and mount commands. Options specified with the mmchfs and mmremoteefs commands are recorded in the GPFS configuration files and are passed as default options to subsequent mount commands on all nodes in the cluster. Options specified with the mmmount or mount commands override the existing default settings and are not persistent.

Unmounting a file system

Some GPFS administration tasks require you to unmount the file system before they can be performed. You can unmount a GPFS file system using the mmumount command.

If the file system does not unmount, see the *File system fails to unmount* section in the *IBM Storage Scale: Problem Determination Guide*.

To unmount a GPFS file system using the mmumount command, enter:

```
mmumount device
```

where *device* is the name of the file system. For example, to unmount the file system **fs1**, enter:

```
mmumount fs1
```

Unmounting a file system on multiple nodes

This topic describes how to unmount a file system on multiple nodes.

To unmount file system **fs1** on all nodes in the GPFS cluster, issue this command:

```
mmumount fs1 -a
```

To unmount a file system only on a specific set of nodes, use the `-N` flag of the `mmumount` command.

Related tasks

Unmounting a file system through GUI

You can use the IBM Storage Scale GUI to mount or unmount individual file systems or multiple file systems on the selected nodes. Use the **Files > File Systems**, **Files > File Systems > View Details > Nodes**, or **Nodes > View Details > File Systems** page in the GUI to mount or unmount a file system.

Unmounting a file system through GUI

You can use the IBM Storage Scale GUI to mount or unmount individual file systems or multiple file systems on the selected nodes. Use the **Files > File Systems**, **Files > File Systems > View Details > Nodes**, or **Nodes > View Details > File Systems** page in the GUI to mount or unmount a file system.

You can utilize the following options that are supported in the GUI to unmount file systems:

- Unmount local file system from local nodes and remote nodes.
- Unmount a remote file system from the local nodes. When a local file system is unmounted from the remote nodes, the remote nodes can no longer be seen in the GUI. The **Files > File Systems > View Details > Remote Nodes** page lists the remote nodes that currently mount the selected file system. The selected file system can be a local or a remote file system but the GUI permits to unmount only local file systems from the remote nodes.
- Select individual nodes, protocol nodes, or nodes by node class while selecting nodes from which the file system needs to be unmounted.
- Specify whether to force unmount. Selecting the **Force unmount option** while unmounting the file system unmounts the file system even if it is still busy in performing the I/O operations. Forcing the unmount operation affects the outstanding operations and causes data integrity issues. The IBM Storage Scale system relies on the native unmount command to carry out the unmount operation. The semantics of forced unmount are platform-specific. On certain platforms such as Linux, even when forced unmount is requested, file system cannot be unmounted if it is still referenced by system kernel. To unmount a file system in such cases, identify and stop the processes that are referencing the file system. You can use system utilities like `lsof` and `fuser` for this.

Related tasks

Unmounting a file system on multiple nodes

This topic describes how to unmount a file system on multiple nodes.

Deleting a file system

Before deleting a file system, unmount it on all nodes.

Specify the file system to be deleted on the `mmdelfs` command. For example, to delete the file system **fs1**, enter:

```
mmdelfs fs1
```

The system displays information similar to:

```
GPFS: 6027-573 All data on the following disks of fs1 will be destroyed:
```

```
gpfs9nsd
gpfs13nsd
gpfs11nsd
gpfs12nsd
```

```
GPFS: 6027-574 Completed deletion of file system fs1.
```

```
mmdelfs: 6027-1371 Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

For more information, see the following:

- [“Unmounting a file system” on page 221](#)

- **`mmdelfs`** command in *IBM Storage Scale: Command and Programming Reference Guide* for complete usage information
- **`mmdelnsd`** command in *IBM Storage Scale: Command and Programming Reference Guide* for removing the NSD definitions after deleting the file system

Determining which nodes have a file system mounted

The **`mmlsmount`** command is used to determine which nodes have a specific file system mounted. The name and IP address of each node, that has the file system mounted, is displayed. This command can be used for all file systems, all remotely mounted file systems, or file systems mounted on nodes of certain clusters.

The **`mmlsmount -L`** command reports file systems that are in use at the time the command is issued. A file system is considered to be in use when it is explicitly mounted with the **`mount`** or **`mmmount`** command, or when it is mounted internally for the purposes of running some other GPFS command. For example, when you run the **`mmrestripefs`** command, the file system remains internally mounted during that time. If the **`mmlsmount`** command is issued in the interim, then the file system is reported as in use by the **`mmlsmount`** command. However, unless the file system is explicitly mounted, it does not show up in the output of the **`mount`** or **`df`** commands. For more information, see *The mmlsmount command* section in the *IBM Storage Scale: Problem Determination Guide*.

This is an example of a **`mmlsmount -L`** command for a mounted file system that is named `fs1`:

```
File system fs1 (mnscd.cluster:fs1) is mounted on 5 nodes:
  9.114.132.101  c5n101          mnscd.cluster
  9.114.132.100  c5n100          mnscd.cluster
  9.114.132.106  c5n106          mnscd.cluster
  9.114.132.97   c5n97           cluster1.cluster
  9.114.132.92   c5n92           cluster1.cluster
```

Checking and repairing a file system

The **`mmfsck`** command detects and repairs conditions that can cause problems in a file system. It operates in two modes: online and offline.

In the online mode the command can run while the file system is still mounted. The command detects and repairs the following conditions:

- Blocks that are marked as allocated but that do not belong to any file (lost blocks). The corrective action is to mark the blocks as free in the block allocation map. A possible symptom of lost blocks is that I/O operations fail with an out-of-space error after repeated node failures.
- Corruptions in the block allocation map. The corrective action is to repair the corruptions.

The command also reports any other problems that it detects.

Note:

- Run the **`mmfsck`** command in the online mode only when the system demand is low. The repairs are I/O-intensive and can degrade system performance.
- If you are repairing a file system because of node failure and the file system has quotas that are enabled, it is a good idea to run the **`mmcheckquota`** command after you run the **`mmfsck`** command to make quota accounting consistent.

In the offline mode the **`mmfsck`** command can run only if the file system is unmounted. In general, you do not need to run the command in offline mode unless you are directed by the IBM Support Center. In the offline mode the command does the same checks that are done in online mode and it also detects and repairs the following problems:

- Blocks marked allocated that do not belong to any file. The corrective action is to mark the block free in the allocation map.
- Files and directories for which an inode is allocated and no directory entry exists, known as orphaned files. The corrective action is to create directory entries for these files in a `lost+found` subdirectory

in the root directory of the fileset to which the file or directory belongs. A fileset is a subtree of a file system namespace that in many respects behaves like an independent file system. The index number of the inode is assigned as the name. If you do not allow the **mmfsck** command to reattach an orphaned file, it asks for permission to delete the file.

- Directory entries that point to an inode that is not allocated. The corrective action is to remove the directory entry.
- Incorrectly formed directory entries. A directory file contains the inode number and the generation number of the file to which it refers. When the generation number in the directory does not match the generation number that is stored in the file's inode, the corrective action is to remove the directory entry.
- Incorrect link counts on files and directories. The corrective action is to update them with accurate counts.
- Policy files that are not valid. The corrective action is to delete the file.
- Various problems that are related to filesets: missing or corrupted fileset metadata, inconsistencies in directory structure related to filesets, missing or corrupted fileset root directory, other problems in internal data structures. The repaired filesets are renamed as *Fileset FilesetId* and put into unlinked state.

The **mmfsck** command performs other functions that are not listed here, as deemed necessary by GPFS.

The **--patch-file** parameter of the **mmfsck** command can be used to generate a report of file system inconsistencies. Consider this example of a patch file that is generated by **mmfsck** for a file system with a bad directory inode:

```
gpfs_fsck

<header>
  sgid = "C0A87ADC:5555C87F"
  disk_data_version = 1
  fs_name = "gpfsh0"
  #patch_file_version = 1
  #start_time = "Fri May 15 16:32:58 2015"
  #fs_manager_node = "h0"
  #fsck_flags = 150994957
</header>

<patch_inode>
  patch_type = "dealloc"
  snapshot_id = 0
  inode_number = 50432
</patch_inode>

<patch_block>
  snapshot_id = 0
  inode_number = 3
  block_num = 0
  indirection_level = 0
  generation_number = 1
  is_clone = false
  is_directory_block = true
  rebuild_block = false
  #num_patches = 1

  <patch_dir>
    entry_offset = 48
    entry_fold_value = 306661480
    delete_entry = true
  </patch_dir>
</patch_block>

<patch_block>
  snapshot_id = 0
  inode_number = 0
  block_num = 0
  indirection_level = 0
  generation_number = 4294967295
  is_clone = false
  is_directory_block = false
  rebuild_block = false
  #num_patches = 1
```

```

<patch_field>
  record_number = 3
  field_id = "inode_num_links"
  new_value = 2
  old_value = 3
</patch_field>
</patch_block>

<patch_inode>
  patch_type = "orphan"
  snapshot_id = 0
  inode_number = 50433
</patch_inode>

<footer>
  #stop_time = "Fri May 15 16:33:06 2015"
  #num_sections = 203
  #fsck_exit_status = 8
  need_full_fsck_scan = false
</footer>

```

The **mmfsck** command can be run with both the **--patch-file** and **--patch** parameters to repair a file system with the information that is stored in the patch file. Using a patch file prevents a subsequent scan of the file system before the repair actions begin.

You cannot run the **mmfsck** command on a file system that has disks in a **down** state. You must first run the mmchdisk command to change the state of the disks to **unrecovered** or **up**. To display the status of the disks in the file system, issue the **mmlsdisk** command.

To check the file system **fs1** without making any changes to the file system, issue the following command:

```
mmfsck fs1
```

For complete usage information, see **mmchdisk command**, **mmcheckquota command**, **mmfsck command**, and **mmlsdisk command** in *IBM Storage Scale: Command and Programming Reference Guide*.

Dynamic validation of descriptors on disk

IBM Storage Scale can periodically scan descriptors on disk to detect and fix corruption early rather than waiting until the next remount.

The first time a file system gets mounted, a periodic validation of the nsd, disk, and stripe group descriptors gets started. This validation occurs, by default, every 5 seconds. The nsd, disk, and stripe group descriptors are read and compared with the corresponding descriptors in memory or cache. If a mismatch occurs, that information is logged and, if appropriate, the file system gets panicked. The recovery steps to fix corrupted data from cache are also logged.

File system maintenance mode

Use file system maintenance mode to enable an IBM Storage Scale file system maintenance window.

Overview

Use file system maintenance mode whenever you perform maintenance on either NSD disks or NSD servers that might result in NSDs becoming unavailable. You cannot change any user files or file system metadata while the file system is in maintenance mode. This way the system does not mark down NSD disks or NSD server nodes when I/O failures occur on those disks because they are not available (because of maintenance). Then, administrators can easily complete administrative actions on the NSD disks or NSD server nodes.

IBM Storage Scale file system operations that must internally mount the file system cannot be used while the file system is in maintenance mode. Other file system administrative operations, such as the operations run by the **mmlsfs** and **mmlsdisk** commands, can check the file system information.

Using file system maintenance mode

You can move the file system into maintenance mode to prevent unexpected or unwanted disk I/O operations in the file system when maintenance actions are applied to either the NSD disk systems or file system server nodes. I/O failures from any NSD disks or server nodes that are not available might result in disks that are marked as down if you do not move the file system into maintenance mode. Any disks that are marked as down must be manually started by using the **mmchdisk** command, which might take significant time for a large file system.

Additionally, no ordering assurance exists for the IBM Storage Scale nodes when you start or shut down nodes across the cluster. So, if the NSD servers are being shut down earlier than client nodes or started up later than client nodes, some NSD disks might also be marked down if I/O operations are run on those NSD server nodes. Unless the file system is in maintenance mode, you must manually control the shutdown or startup sequence for cluster nodes to avoid disk down events.

You can move the file system into maintenance mode before you shut down or mount the file system during the start process. Do this to release the control on the orders of nodes shutdown or startup sequence. When you remotely mount and access a file system, you should move the file system into maintenance mode before you shut down the NSD servers in the home cluster. Do this because users of remote file system might be not aware of the home cluster status. Then initiating I/O operations from remote cluster might cause file system disks to be marked down as well.

Setting up file system maintenance mode

You can enable, disable, or check the status of file system maintenance mode:

- To enable or disable file system maintenance mode, enter the following command:

```
mmchfs <fsName> --maintenance-mode yes [-wait] | no
```

- To check the status of file system maintenance mode, enter the following command:

```
mmlsfs <fsName> --maintenance-mode
```

Before you enter the **mmchfs** command to enable file system maintenance mode, make sure that you unmount the file system on the local and remote clusters. Additionally, long running commands such as **mmrestripefs** must complete because they internally mount the file system. If you cannot wait for long running commands, you must specify the **--wait** parameter. The **--wait** parameter waits on existing mounts and long running commands, and moves the file system into maintenance mode after all existing mounts and long running commands complete.

You can apply maintenance on network shared disk (NSD) disks or server nodes:

1. Unmount the file system from all nodes, including remote cluster nodes. Enter the following command:

```
mmumount <fsName> -a
```

2. Check whether any pending internal mounts exist. Enter the following command:

```
mmlsmount <fsName> -L
```

3. Enter the following command to enable maintenance mode:

```
mmchfs <fsName> --maintenance-mode yes
```

Remember: If you use the **--wait** parameter with the **mmchfs** command, file system maintenance mode is enabled automatically after you unmount the file system from all local and remote nodes.

4. Complete any needed maintenance on the NSDs or server nodes. Maintenance tasks on NSDs or server nodes include these tasks:

- You can restart the NSD servers.
- You can stop any access to NSDs.

- You can shut down the entire cluster safely when the file system is in maintenance mode.

Note: File system mount and other management operations that internally mount file system cannot run in this state, such as **mmmount** and **mmrestripefs**:

```
mmmount <fsName>
Mon Jul 23 06:02:49 EDT 2018: 6027-1623
mmmount: Mounting file systems ...
mount: permission denied
mmmount: 6027-1639 Command failed. Examine previous error messages to determine cause.
```

```
mmrestripefs <fsName> -b
This file system is undergoing maintenance and cannot be either mounted or changed.
mmrestripefs: 6027-1639 Command failed. Examine previous error messages to determine cause.
```

5. Resume the normal file system operations such as **mmmount** after maintenance is complete. End the maintenance mode only after the NSD disks and NSD servers are operational:

```
mmchfs <fsName> --maintenance no
```

You can run offline **fsck** to check file system consistency before you resume file system maintenance mode.



CAUTION:

- If you shut down either the NSD servers or the whole cluster, it is considered maintenance on NSD disks or servers and must be done under maintenance mode.
- If no NSD disks or NSD server nodes are available for a specified file system, the file system maintenance mode state cannot be retrieved because it is stored with the stripe group descriptor. Additionally, you cannot resume the file system maintenance mode in this scenario.

Running the **fsck** service action while the file system is in maintenance mode

The offline **fsck** service action can be run while the file system is in maintenance mode. Maintenance mode is used to provide a dedicated timing window to check file system consistency when:

- The offline **fsck** service action cannot be started while the file system is being used.
- The offline **fsck** service action cannot be started due to some unexpected interfering file system mount or other management operations.

Do not specify these commands if your file system is in maintenance mode:

- **mmmount**
- **mmrestripefs**
- **mmdelfs**
- **mmdefragfs**
- **mmadddisk**
- **mmdeletdisk**
- **mmrpldisk**
- **mmchdisk**
- **mmcrsnapshot**
- **mmdeletesnapshot**
- **mmcrfileset**
- **mmdelfileset**
- **mmchfileset**
- **mmchqos**
- **mmchpolicy**

- **mmquotaon**
- **mmquotaooff**
- **mmedquota**
- **mmdefedquota**
- **mmdefquotaon**
- **mmdefquotaoff**
- **mmsanrepairfs**
- **mmputacl**

Note: These commands fail when you specify them while your file system is in maintenance mode.

See also

- *mmchfs*
- *mmlsfs*

Listing file system attributes

Use the **mmlsfs** command to display the current file system attributes. Depending on your configuration, additional information that is set by GPFS can be displayed to help in problem determination when you contact the IBM Support Center.

If you specify no options with the **mmlsfs** command, all file system attributes are listed.

For example, to list all of the attributes for the file system `gpfs1`, enter:

```
mmlsfs gpfs1
```

The system displays information similar to the following:

flag	value	description
-f	8192	Minimum fragment (subblock) size in bytes
-i	512	Inode size in bytes
-I	16384	Indirect block size in bytes
-m	1	Default number of metadata replicas
-M	2	Maximum number of metadata replicas
-r	1	Default number of data replicas
-R	2	Maximum number of data replicas
-j	cluster	Block allocation type
-D	nfs4	File locking semantics in effect
-k	all	ACL semantics in effect
-n	32	Estimated number of nodes that will
mount file system		
-B	262144	Block size
-Q	none	Quotas accounting enabled
	none	Quotas enforced
	none	Default quotas enabled
--perfileset-quota	no	Per-fileset quota enforcement
--filesetdf	no	Fileset df enabled?
-V	25.00 (5.1.1.0)	File system version
--create-time	Mon Jul 4 22:20:38 2022	File system creation time
-z	yes	Is DAPI enabled?
-L	4194304	LogFile size
-E	yes	Exact mtime mount option
-S	relatime	Suppress atime mount option
-K	whenpossible	Strict replica allocation option
--fastea	yes	Fast external attributes enabled?
--encryption	no	Encryption enabled?
--inode-limit	879616	Maximum number of inodes in all inode
spaces		
--log-replicas	0	Number of log replicas
--is4KAigned	no	is4KAigned?
--rapid-repair	yes	rapidRepair enabled?
--write-cache-threshold	0	HAWC Threshold (max 65536)
--subblocks-per-full-block	32	Number of subblocks per full block
-P	system	Disk storage pools in file system
--file-audit-log	no	File Audit Logging enabled?

--maintenance-mode	no	Maintenance Mode enabled?
--flush-on-close	no	flush cache on file close enabled?
--auto-inode-limit	no	Increase maximum number of inodes per
inode space automatically?		
--nfs4-owner-write-acl	yes	NFSv4 implicit owner WRITE_ACL
permission enabled?		
-d	nsd9;nsd10	Disks in file system
-A	yes	Automatic mount option
-o	none	Additional mount options
-T	/gpfs/gpfs2	Default mount point
--mount-priority	0	Mount priority

Some of the attributes that are displayed by the `mmlsfs` command represent default mount options. Because the scope of mount options is an individual node, it is possible to have different values on different nodes. For exact `mtime` (-E option) and suppressed `atime` (-S option), the information that is displayed by the `mmlsfs` command represents the current setting on the file system manager node. If these options are changed with the `mmchfs` command, the change might not be reflected until the file system is remounted.

For complete usage information, see **`mmlsfs` command** in *IBM Storage Scale: Command and Programming Reference Guide*. For a detailed discussion of file system attributes, see *GPFS architecture* and *File system creation considerations* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Modifying file system attributes

Use the `mmchfs` command to modify existing file system attributes.

Note: All files created after issuing the `mmchfs` command take on the new attributes. Existing files are not affected. Use the `mmchattr` or `mmrestripefs -R` command to change the replication factor of existing files. See “[Querying and changing file replication attributes](#)” on page 229.

For example, to change the default data replication factor to 2 for the file system **fs1**, enter:

```
mmchfs fs1 -r 2
```

To confirm the changes, enter:

```
mmlsfs fs1 -r
```

The system displays information similar to:

flag	value	description
-r	2	Default number of data replicas

For complete usage information, see **`mmchfs` command** and **`mmlsfs` command** in *IBM Storage Scale: Command and Programming Reference Guide*. For a detailed discussion of file system attributes, see *GPFS architecture* and *File system creation considerations* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Querying and changing file replication attributes

If your availability requirements change, you can have GPFS display the current replication factors for one or more files by issuing the `mmlsattr` command. You might then decide to change replication for one or more files using the `mmchattr` command.

For complete usage information, see **`mmlsattr` command** and **`mmchattr` command** in *IBM Storage Scale: Command and Programming Reference Guide*.

Querying file replication

Specify one or more file names with the `mmlsattr` command.

For example, to display the replication factors for two files named `project4.sched` and `project4.resource` in the file system **fs1**, enter:

```
mmlsattr /fs1/project4.sched /fs1/project4.resource
```

The system displays information similar to:

```
replication factors
metadata(max) data(max) file      [flags]
-----
1 ( 2)   1 ( 2) /fs1/project4.sched
1 ( 2)   1 ( 2) /fs1/project4.resource
```

See the **`mmlsattr` command** in *IBM Storage Scale: Command and Programming Reference Guide* for complete usage information. For a detailed discussion of file system attributes, see *GPFS architecture and File system creation considerations* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Related tasks

[Changing file replication attributes](#)

Use the `mmchattr` command to change the replication attributes for one or more files.

Changing file replication attributes

Use the `mmchattr` command to change the replication attributes for one or more files.

You can only increase data and metadata replication as high as the maximum data and maximum metadata replication factors for that file system. You cannot change the maximum data and maximum metadata replication factors once the file system has been created.

Specify the file name, attribute, and new value with the `mmchattr` command. For example, to change the metadata replication factor to 2 and the data replication factor to 2 for the file named `project7.resource` in the file system **fs1**, enter:

```
mmchattr -m 2 -r 2 /fs1/project7.resource
```

To confirm the change, enter:

```
mmlsattr /fs1/project7.resource
```

The system displays information similar to:

```
replication factors
metadata(max) data(max) file      [flags]
-----
2 ( 2)   2 ( 2)   /fs1/project7.resource
```

See the **`mmchattr` command** and the **`mmlsattr` command** in *IBM Storage Scale: Command and Programming Reference Guide* for complete usage information. For a detailed discussion of file system attributes, see *GPFS architecture and File system creation considerations* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Related tasks

[Querying file replication](#)

Specify one or more file names with the `mm1sattr` command.

Using Direct I/O on a file in a GPFS file system

The Direct I/O caching policy can be set for files in a GPFS file system by specifying the -D option on the `mmchattr` command.

This caching policy bypasses file cache and transfers data directly from disk into the user space buffer, as opposed to using the normal cache policy of placing pages in kernel memory. Applications with poor cache hit rates or very large I/Os may benefit from the use of Direct I/O.

Direct I/O may also be specified by supplying the O_DIRECT file access mode on the open() of the file.

File compression

You can compress or decompress files either with the `mmchattr` command or with the `mmapplypolicy` command with a **MIGRATE** rule. With the MIGRATE rule, administrators can create policies that select a compression library based on the access characteristics of the file to be compressed, with file-level granularity. You can do the compression or decompression synchronously or defer it until a later call to `mmrestripefile` or `mmrestripefs`.

The supported compression libraries are z, lz4, zfast, alphae, and alphah. They are intended primarily for compressing the following types of data:

z

Cold data. Favors compression efficiency over access speed.

lz4

Active, non-specific data. Favors access speed over compression efficiency.

zfast

Active genomic data in FASTA, SAM, or VCF format.

alphae

Active genomic data in FASTQ format. Slightly favors compression efficiency over access speed.

alphah

Active genomic data in FASTQ format. Slightly favors access speed over compression efficiency.

The following table shows the IBM Storage Scale file system format level that is required for each compression library.

Table 17. Compression libraries and their required file system format level and format number	
Compression library	Required file system format level and format number
z	4.2.0 (15.01) or later
lz4	5.0.0 (18.00) or later
zfast, alphae, alphah	5.0.3 (21.00) or later

For more information about file compression, see the following subtopics:

- “Comparison with object compression” on page 232
- “Setting up file compression and decompression” on page 232
- “Warning” on page 233
- “Reported size of compressed files” on page 233
- “Deferred file compression” on page 233
- “Indicators of file compression or decompression” on page 234
- “Partially compressed files” on page 235
- “Updates to compressed files” on page 235

- “File compression and memory mapping” on page 236
- “File compression and direct I/O” on page 236
- “Backing up and restoring compressed files” on page 236
- “FPO environment” on page 236
- “AFM environment” on page 236
- “Limitations” on page 237

Comparison with object compression

File compression and object compression use the same compression technology but are available in different environments and are configured in different ways. Object compression is available in the Cluster Export Systems (CES) environment and is configured with the **mmobj policy** command. With object compression, you can create an object storage policy that periodically compresses new objects and files in a GPFS file set.

File compression is available in non-CES environments and is configured with the **mmapplypolicy** command or directly with the **mmchattr** command.

Setting up file compression and decompression

The sample script `/usr/lpp/mmfs/samples/ilm/mmcompress.sample`, installed with IBM Storage Scale, provides examples of how to compress or decompress a file set or a directory tree.

You can do file compression or decompression with either the **mmchattr** command or the **mmapplypolicy** command.

With the **mmchattr** command, you specify the **--compression** option and the names of the files or file sets that you want to compress or decompress. For example,

- The following command compresses a file with the lz4 compression library:

```
mmchattr --compression lz4 trcpt.150913.13.30.13.3518.txt
```

- The following command decompresses the same file:

```
mmchattr --compression no trcpt.150913.13.30.13.3518.txt
```

For more information, see the topic *mmchattr command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

With the **mmapplypolicy** command, you create a **MIGRATE** rule that specifies the **COMPRESS** option and run **mmapplypolicy** to apply the rule.

Note: File compression and decompression with the **mmapplypolicy** command is not supported on Windows.

See the following examples:

- The following rule selects files with names that contain the string **green** from the **datapool** storage pool and compresses them with the z library:

```
RULE 'COMPR1' MIGRATE FROM POOL 'datapool' COMPRESS('z') WHERE NAME LIKE 'green%'
```

- The following rule decompresses the same set of files:

```
RULE 'COMPR1' MIGRATE FROM POOL 'datapool' COMPRESS('no') WHERE NAME LIKE 'green%'
```

- The following example shows three rules:

- The first rule excludes from compression any file that ends with **.mpg** or **.jpg**.
- The second rule automatically compresses any file that was not accessed in the last 30 days with z (**libz.so**).

- The third rule automatically compresses any file that was not modified in the last 2 days with lz4 (liblz4.so).

```
RULE 'NEVER_COMPRESS' EXCLUDE WHERE lower(NAME) LIKE '%.mpg' OR lower(NAME) LIKE '%.jpg'
RULE 'COMPRESS_COLD' MIGRATE COMPRESS('z') WHERE (CURRENT_TIMESTAMP - ACCESS_TIME) >
(INTERVAL '30' DAYS)
RULE 'COMPRESS_ACTIVE' MIGRATE COMPRESS('lz4') WHERE (CURRENT_TIMESTAMP - MODIFICATION_TIME) >
(INTERVAL '2' DAYS) AND (CURRENT_TIMESTAMP - ACCESS_TIME) <= (INTERVAL '30' DAYS)
```

- The following rule compresses genomic data in files with the extensions .fastq and .fq:

```
RULE 'COMPRESS_GENOMIC' MIGRATE COMPRESS('alphae') WHERE lower(NAME) LIKE '%.fastq' OR lower(NAME) LIKE '%.fq'
```

For more information, see the following help topics:

- The topic ***mmchattr command*** in the *IBM Storage Scale: Command and Programming Reference Guide*.
- “[Overview of policies](#)” on page 531
- “[Policy rules: Syntax](#)” on page 534
- “[Policy rules: Terms](#)” on page 535

When you do file compression, you can defer the compression operation to a later time. For more information, see the subtopic “[Deferred file compression](#)” on page 233.

Warning

Doing any of the following operations while the **mmrestorefs** command is running can corrupt file data:

- Doing file compression or decompression. This includes compression or decompression with the **mmchattr** command or with a policy and the **mmapplypolicy** command.
- Running the **mmrestripefile** command or the **mmrestripefs** command. Do not run either of these commands for any reason. Do not run these commands to complete a deferred file compression or decompression.

Reported size of compressed files

After a file is compressed, operating system commands, such as **ls -l**, display the uncompressed size. Use **du** or the GPFS command **mmdf** to display the actual, compressed size. You can also make the **stat()** system call to find how many blocks the file occupies.

Deferred file compression

By default, the command that starts a file compression or decompression does not return until after the compression or decompression operation is completed. However, with both the **mmchattr** command and the **mmapplypolicy** compression, you can defer the compression or decompression operation and have the command return as soon as it completes any other operations. By deferring compression or decompression, you can complete the operation later when the system is not heavily loaded with processes or I/O.

To defer the compression, with either command, specify the **-I defer** option. For example, the following command marks the specified file as needing compression but defers the compression operation:

```
mmchattr -I defer --compression yes trcrpt.150913.13.30.13.3518.txt
```

With the **mmapplypolicy** command, the **-I defer** option defers compression or decompression and data movement or deletion. For example, the following command applies the rules in the file **policyfile** but defers the file operations that are specified in the rules, including compression or decompression:

```
mmapplypolicy fs1 -P policyfile -I defer
```

To complete a deferred compression or decompression, run the **mmrestripefile** command or the **mmrestripefs** command with the -z option. (Do not run either of these commands if an **mmrestorefs** command is running. See the warnings in the preceding subtopic “[Warning](#)” on page 233.) The following command completes the deferred compression or decompression of the specified file:

```
mmrestripefile -z trcpt.150913.13.30.13.3518.txt
```

Indicators of file compression or decompression

The **mmlsattr** command displays two indicators that together describe the state of compression or decompression of the specified file:

COMPRESSION

The **mmlsattr** command displays the COMPRESSION flag on the Misc attributes line of its output. The flag is followed in parentheses by the name of the compression library that was used to compress the file. See the example of **mmlsattr** output in [Figure 9 on page 235](#). If present, the COMPRESSION flag indicates that the file is compressed or is marked for deferred compression. If absent, the absence indicates that the file is uncompressed or is marked for deferred decompression.

Note: This flag reflects the state of the GPFS_IWINFLAG_COMPRESSED flag in the **gpfs_iattr64_t** structure of the inode of the file. For more information about this structure, see the topic *gpfs_iattr64_t_structure* in the *IBM Storage Scale: Command and Programming Reference Guide*.

illCompressed

The **mmlsattr** command displays the illCompressed flag on the flags line of its output. See [Figure 9 on page 235](#). If present, illCompressed indicates that the file is marked for compression or decompression but that compression or decompression is not completed. If absent, the absence indicates that compression or decompression is completed. For more information about this structure, see the topic *gpfs_iattr64_t_structure* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Note:

- This flag reflects the state of the GPFS_IAFLAG_ILLCOMPRESSED flag in the **gpfs_iattr64_t** structure of the inode of the file. For more information about this structure, see the topic *gpfs_iattr64_t_structure* in the *IBM Storage Scale: Command and Programming Reference Guide*.
- Some file system events can cause the illCompressed flag to be set. Consider the following examples:
 - When data is written into an already compressed file, the existing data remains compressed but the new data is uncompressed. The illCompressed flag is set for this file.
 - When a compressed file is memory-mapped, the memory-mapped area of the file is decompressed before it is read into memory. The illCompressed flag is set for this file.

For more information, see the subtopic “[Updates to compressed files](#)” on page 235.

In the following example, the output from the **mmlsattr** command includes both the COMPRESSION flag and the illCompressed flag. This combination indicates that the file is marked for compression but that compression is not completed:

```
mmlsattr -L green02.51422500687
file name:           green02.51422500687
metadata replication: 1 max 2
data replication:   2 max 2
immutable:          no
appendOnly:         no
flags:              illCompressed
storage pool name: datapool
fileset name:       root
snapshot name:
creation time:     Wed Jan 28 19:05:45 2015
Misc attributes:   ARCHIVE COMPRESSION (library lz4)
Encrypted:         no
```

Figure 9. Compression and decompression flags

Together the COMPRESSION and illCompressed flags indicate the compressed or uncompressed state of the file. See the following table.

Table 18. COMPRESSION and illCompressed flags		
State of the file	COMPRESSION is displayed?	illCompressed is displayed?
Uncompressed.	No	No
Decompression is not complete.	No	Yes
Compressed.	Yes	No
Compression is not complete.	Yes	Yes

Partially compressed files

The COMPRESSION flag of a file is set when the user selects the file to be compressed by the **mmchattr --compress yes** command or by a policy run. The flag indicates that the user wants the file to be compressed.

If the user specifies the **-I defer** command option with the **mmchattr** command or with a policy run, the illCompressed flag of the file is set during the command execution or policy run. The illCompressed flag indicates that the request to compress the file has not been fulfilled. The illCompressed flag is reset at the conclusion of the actual compression execution of the file, after the **mmrestripefs -z** or **mmrestripefile -z** command finishes compressing the file. The illCompressed flag can be set again upon updates of the contents of the file that cause update-driven decompression.

The compressibility of a file can change over time if its contents are changed. Different parts of a file might have different compressibility. Based on the 10% space-saving criterion (see the subtopic “[Limitations](#)” on page 237), some compression groups (in granularity of 10 data blocks) of a file might be compressed while others are not.

In sum, the state of the COMPRESSION flag, on or off, indicates the intention of the user to compress the file or not. The illCompressed flag indicates the compression execution status. The actual compression status of the data blocks depends on the illCompressed and COMPRESSION flags and the compressibility of the current data.

Updates to compressed files

When a compressed file is updated by a write operation, the file system automatically decompresses the region of the file that contains the affected data and sets the illCompressed flag. The file system then makes the update. To recompress the file, run the **mmrestripefile** command with the **-z** option, as in the following example:

```
mmrestripefile -z trcrpt.150913.13.30.13.3518.txt
```

The **mmrestorefs** command can cause a compressed file in the active file system to become decompressed if it is overwritten by the restore process. To re-compress the file, run the **mmrestripefile** command with the **-z** option.

For more information, see the preceding subtopic “[Deferred file compression](#)” on page 233.

File compression and memory mapping

On Linux and AIX you can memory-map a file that is already compressed. The file system automatically decompresses the paged-in region and sets the `illCompressed` flag. To re-compress the file, run the **mmrestripefile** command with the **-z** option.

As a convenience, the file system does not compress an uncompressed file or partially decompressed file if the file is memory-mapped. Compressing the file would not be effective because memory mapping decompresses any compressed data in the regions that are paged in.

File compression and direct I/O

You can open a compressed file for Direct I/O, but internally the direct I/O reads and writes are replaced by buffered decompressed I/O reads and writes.

As a convenience, the file system does not compress a file that is opened for Direct I/O. Compressing the file would not be effective because direct I/O would be replaced by buffered decompressed I/O.

Backing up and restoring compressed files

Files are decompressed when they are moved out of storage that is directly managed by IBM Storage Scale. This fact affects file backups by products such as IBM Storage Protect, IBM Storage Protect for Space Management, IBM Spectrum Archive, Transparent cloud tiering (TCT), and others. When you back up a file with these products, the file system decompresses the file data inline when it is read by the backup agent. The file system also sets the `illCompressed` flag in the file properties. The backed-up file data is not compressed.

When you restore a file to the IBM Storage Scale file system, the file data remains uncompressed but the `illCompressed` flag is still set. You can re-compress the file by running **mmrestorefs** or **mmrestripefile** with the **-z** option.

FPO environment

File compression supports a File Placement Optimizer (FPO) environment or horizontal storage pools.

FPO block group factor: Before you compress files in a File Placement Optimizer (FPO) environment, you must set the block group factor to a multiple of 10. If you do not, then data block locality is not preserved and performance is slower.

For compatibility reasons, before you do file compression with FPO files, you must upgrade the whole cluster to version 4.2.1 or later. To verify that the cluster is upgraded, follow these steps:

1. At the command line, enter the **mm1sconfig** command with no parameters.
2. In the output, verify that `minReleaseLevel` is 4.2.1 or later.

AFM environment

Files that belong to AFM and AFM DR filesets can also be compressed and decompressed. Compressed file contents are decompressed before being transferred from home to cache or from primary to secondary.

Before you do file compression with AFM and AFM DR, you must upgrade the whole cluster to version 5.0.0.

Limitations

See the restrictions that are stated in the following subtopics:

- [“File compression and memory mapping” on page 236](#)
- [“File compression and direct I/O” on page 236](#)
- [“Backing up and restoring compressed files” on page 236](#)

File compression also has the following limitations:

- File compression processes each compression group within a file independently. A compression group consists of one to 10 consecutive data blocks within a file. If the file contains fewer than 10 data blocks, then the whole file is one compression group. If the saving of space for a compression group is less than 10%, then the file compression does not compress it, but skips to the next compression group.
- For file-enabled compression in an FPO-enabled file system, the block group factor must be a multiple of 10, so that the compressed data maintains data locality. If the block group factor is not a multiple of 10, then the data locality is broken.
- Direct I/O is not supported for compressed files.
- File compression is not supported on a file system where HAWC is enabled.
- The following operations are not supported:
 - Compressing files in snapshots
 - Compressing a clone
 - Compressing small files (files that occupy fewer than two subblocks, compressing small files into an inode).
 - Compressing files other than regular files, such as directories.
 - Cloning a compressed file
 - Compressing an open file that is memory-mapped. See the subtopic [“File compression and memory mapping” on page 236](#).
- Additional limitations on Windows:
 - Compression or decompression with the **mmapplypolicy** command is not supported.
 - Compression of files in Windows hyper allocation mode is not supported.
 - Memory mapping a file that is already compressed is not supported.
 - The following Windows APIs are not supported:
 - FSCTL_SET_COMPRESSION to enable/disable compression on a file.
 - FSCTL_GET_COMPRESSION to retrieve compression status of a file.
 - In Windows Explorer, in the **Advanced Attributes** window, the compression feature is not supported.

Setting the Quality of Service for I/O operations

QoS limits the effect of I/O-intensive GPFS maintenance commands on overall system I/O performance.

With QoS, you can prevent I/O-intensive, long-running GPFS maintenance commands from dominating file system I/O performance and significantly delaying other tasks. Commands like the examples in [Examples of long-running, IO-intensive GPFS commands](#) can generate hundreds or thousands of requests for I/O operations per second. The high demand can greatly slow down normal tasks that are competing for the same I/O resources.

```
mmrestripefs fname -N  
mmapplypolicy fname -N all ...
```

The I/O intensive, potentially long-running GPFS commands are collectively called *maintenance commands* and are listed in the help topic for the *mmchqos* command in the *IBM Storage Scale: Command and Programming Reference Guide*.

With QoS configured, you can assign an instance of a maintenance command to a QoS class that has a lower I/O priority. Although the instance now takes longer to run to completion, normal tasks have greater access to I/O resources and run more quickly.

For more information, see the descriptions of the QoS commands:

- *mmchqos command* in the *IBM Storage Scale: Command and Programming Reference Guide*
- *mmlsqos command* in the *IBM Storage Scale: Command and Programming Reference Guide*

Note:

- QoS requires the file system to be at V4.2.0.0 or later. To check the file system level, enter the following command:

```
mmlsfs fileSystemName -V
```

- QoS works with asynchronous I/O, memory-mapped I/O, cached I/O, and buffered I/O. However, with direct I/O, QoS counts the IOPS but does not regulate them.

Overview of using QoS

The following steps provide an overview of how to use QoS. In this overview, assume that the file system `fs0` contains 5 nodes and has two storage pools: the system storage pool (`system`) and another storage pool `sp1`.

1. Monitor your file system with the **`mmlsqos`** command to determine its maximum capacity in I/O operations per second (IOPS). Follow these steps:
 - a. Enable QoS without placing any limits on I/O consumption. The following command sets the QoS classes of both storage pools to unlimited:

Table 19. Set QoS classes to unlimited

Storage pool	QoS class: maintenance	QoS class: other
system	unlimited	unlimited
sp1	unlimited	unlimited

```
mmchqos fs0 --enable --reset
```

- b. Run some maintenance commands that drive I/O on all nodes and disks.
- c. Run the **`mmlsqos`** command to observe how many IOPS are consumed:

```
mmlsqos fs0 --seconds 60
```

2. Run the **`mmchqos`** command to allocate the available IOPS among the storage pools.

- a. Allocate a smaller share of IOPS to the maintenance class, perhaps 15 percent. For example, if you determined in Step 1 that the maximum is 10,000 IOPS, then you might allocate 1500 IOPS to the maintenance class.

If there is more than one storage pool, then divide the IOPS among the maintenance classes of the storage pools. In this overview, suppose that you decide to allocate 1000 IOPS to the maintenance class of the system pool and 500 IOPS to the maintenance class of the `sp1` storage pool. See the second column of the table below.

Note: Make sure that the virtual storage Logical Unit Numbers (LUNs) of different storage pools do not map to the same physical devices.

By default, QoS divides specific allocations of IOPS evenly among the nodes in the file system. In this overview there are 5 nodes. So QoS allocates 200 IOPS to the maintenance class of the system pool and 100 IOPS to the maintenance class of the `sp1` storage pool on each node.

Note: You can also divide IOPS among a list of nodes or among the nodes of a node class. For example, you can use the `mmcrnodeclass` command to create a class of nodes that do maintenance commands. You can then divide IOPS among the members of the node class by entering a command like the following one:

```
mmchqos fs0 --enable -N nodeClass pool=sp2,maintenance=880IOPS,other=unlimited
```

If the file system serves remote clusters, you can divide IOPS among the members of a remote cluster by entering a command like the following one:

```
mmchqos fs0 --enable -C remoteCluster pool=sp3,maintenance=1000IOPS,other=unlimited
```

- b. Allocate the remaining IOPS to the other classes. It is a good idea to accomplish this task by setting `other` to `unlimited` in each storage class. Then normal tasks can absorb all the IOPS of the system when no maintenance commands are running. See the third column of the following table.

Table 20. Allocate the available IOPS

Storage pool	QoS class: maintenance	QoS class: other
system	1000 IOPS (200 IOPS per node)	unlimited
sp1	500 IOPS (100 IOPS per node)	unlimited

The command is on one line:

```
mmchqos fs0 --enable pool=system,maintenance=1000IOPS,other=unlimited
pool=sp1,maintenance=500IOPS,other=unlimited
```

- 3. When you run a maintenance command, QoS by default assigns it to the maintenance class:

```
mmdeldisk fs0 nsd12
```

All maintenance command instances that are running at the same time and that access the same storage pool compete for the IOPS that you allocated to the maintenance class of that storage pool. If the IOPS limit of the class is exceeded, then QoS queues the extra I/O requests until more IOPS become available.

To run a maintenance command without I/O restrictions, you can explicitly assign it to the `other` class:

```
mmdeldisk fs0 nsd12 --qos other
```

- 4. You can disable QoS at any time without losing your IOPS allocations:

```
mmchqos fs0 --disable
```

When you re-enable QoS it starts applying the allocations again:

```
mmchqos fs0 --enable
```

- 5. You can change the IOPS allocations at any time. Execute the following command in one line:

```
mmchqos fs0 --enable pool=system,maintenance=750IOPS,other=unlimited
pool=sp1,maintenance=750IOPS,other=unlimited
```

When you change allocations, mount the file system, or re-enable QoS, a brief delay due to reconfiguration occurs before QoS starts applying allocations.

6. To monitor the consumption of IOPS while a maintenance command is running, run the `mmlsqos` command. The following command displays the statistics for the preceding 60 seconds during which a maintenance command was running:

```
mmlsqos fs0 --seconds 60
```

See also

- *mmchqos command in the IBM Storage Scale: Command and Programming Reference Guide*
- *mmlsqos command in the IBM Storage Scale: Command and Programming Reference Guide*

Restriping a GPFS file system

Writing data into a GPFS file system correctly stripes the file. However, if you have added disks to a GPFS file system that are seldom updated, use the `mmrestripefs` command to restripe the file system to achieve maximum performance. You can also use `mmrestripefs` to perform any incomplete or deferred file compression or decompression.

Restriping offers the opportunity to specify useful options in addition to rebalancing (-b option). Re-replicating (-r or -R option) provides for proper replication of all data and metadata. If you use replication, this option is useful to protect against additional failures after losing a disk. For example, if you use a replication factor of 2 and one of your disks fails, only a single copy of the data would remain. If another disk then failed before the first failed disk was replaced, some data might be lost. If you expect delays in replacing the failed disk, you could protect against data loss by suspending the failed disk using the `mmchdisk` command and re-replicating. This would assure that all data existed in two copies on operational disks.

If files are assigned to one storage pool, but with data in a different pool, the placement (-p) option will migrate their data to the correct pool. Such files are referred to as ill-placed. Utilities, such as the `mmchattr` command or policy engine, may change a file's storage pool assignment, but not move the data. The `mmrestripefs` command may then be invoked to migrate all of the data at once, rather than migrating each file individually. Note that the rebalance (-b) option also performs data placement on all files, whereas the placement (-p) option rebalances only the files that it moves.

If you do not replicate all of your files, the migrate (-m) option is useful to protect against data loss when you have an advance warning that a disk may be about to fail, for example, when the error logs show an excessive number of I/O errors on a disk. Suspending the disk and issuing the `mmrestripefs` command with the -m option is the quickest way to migrate only the data that would be lost if the disk failed.

If you do not use replication, the -m and -r options are equivalent; their behavior differs only on replicated files. After a successful re-replicate (-r option) all suspended disks are empty. A migrate operation, using the -m option, leaves data on a suspended disk as long as at least one other replica of the data remains on a disk that is not suspended. Restriping a file system includes re-replicating it; the -b option performs all the operations of the -m and -r options.

Use the -z option to perform any deferred or incomplete compression or decompression of files in the file system.

Consider the necessity of restriping and the current demands on the system. New data which is added to the file system is correctly striped. Restriping a large file system requires extensive data copying and may affect system performance. Plan to perform this task when system demand is low.

If you are sure you want to proceed with the restripe operation:

1. Use the `mmchdisk` command to suspend any disks to which you *do not* want the file system restriped. You may want to exclude disks from file system restriping because they are failing. See “[Changing GPFS disk states and parameters](#)” on page 281.
2. Use the `mmlsdisk` command to assure that all disk devices to which you *do* want the file system restriped are in the up/normal state. See “[Displaying GPFS disk states](#)” on page 280.

Specify the target file system with the **mmrestripefs** command. For example, to rebalance (-b option) file system **fs1** after adding an additional RAID device, enter:

```
mmrestripefs fs1 -b
```

The system displays information similar to:

```
Scanning file system metadata, phase 1 ...
 19 % complete on Wed Mar 14 21:28:46 2012
 100 % complete on Wed Mar 14 21:28:48 2012
Scan completed successfully.
Scanning file system metadata, phase 2 ...
Scanning file system metadata for sp1 storage pool
Scan completed successfully.
Scanning file system metadata, phase 3 ...
Scan completed successfully.
Scanning file system metadata, phase 4 ...
Scan completed successfully.
Scanning user file metadata ...
 100.00 % complete on Wed Mar 14 21:28:55 2012
Scan completed successfully.
```

Note: Rebalancing of files is an I/O-intensive and time-consuming operation, and is important only for file systems with large files that are mostly invariant. In many cases, normal file update and creation will rebalance your file system over time, without the cost of the rebalancing.

For complete usage information, see **mmrestripefs command** in *IBM Storage Scale: Command and Programming Reference Guide*.

Querying file system space

Although you can use the **df** command to summarize the amount of free space on all GPFS disks, the **mmdf** command is useful for determining how well-balanced the file system is across your disks. (Also, the output from **mmdf** can be more up to date than the output from **df**.) Additionally, you can use the **mmdf** command to diagnose space problems that might result from fragmentation.

Note: The **mmdf** command may require considerable metadata I/O, and should be run when the system load is light.

Specify the file system you want to query with the **mmdf** command. For example, to query available space on all disks in the file system **fs1**, enter:

```
mmdf fs1
```

The system displays information similar to:

disk name	disk size in KB	failure group	holds metadata	holds data	free KB in full blocks	free KB in fragments

Disks in storage pool: system	(Maximum disk size allowed is 122 GB)					
hd16vsdn10	17793024	-1	yes	yes	17538560 (99%)	1728 (0%)
hd3vsdn01	8880128	2	yes	yes	8658176 (98%)	1600 (0%)
hd4vsdn01	8880128	2	yes	yes	8616448 (97%)	1384 (0%)
hd15vsdn10	17793024	10	yes	yes	17539584 (99%)	1664 (0%)
hd13vsdn02	8880128	4001	yes	yes	8663552 (98%)	1776 (0%)
hd8vsdn01	8880128	4002	yes	yes	8659200 (98%)	1936 (0%)
hd5vsdn01	8880128	4002	yes	yes	8654848 (97%)	1728 (0%)
hd33n09	17796008	4003	yes	yes	17540864 (99%)	2240 (0%)
(pool total)	257800488				252091136 (98%)	46928 (0%)

Disks in storage pool: fs1sp1	(Maximum disk size allowed is 122 GB)					
hd30n01	8897968	8	no	yes	8895488 (100%)	424 (0%)
hd31n01	8897968	8	no	yes	8895488 (100%)	424 (0%)
(pool total)	17795936				17790976 (100%)	848 (0%)
=====						
(data)	266716296				261222144 (98%)	44576 (0%)
(metadata)	248920360				243217408 (98%)	46048 (0%)
(total)	275596424				269882112 (98%)	47776 (0%)

```

inode Information
-----
Number of used inodes:      9799
Number of free inodes:      4990393
Number of allocated inodes: 5000192
Maximum number of inodes:   5000192

```

For complete usage information, see **`mmdf` command** in *IBM Storage Scale: Command and Programming Reference Guide*.

Querying and reducing file system fragmentation

Disk fragmentation within a file system is an unavoidable condition. When a file is closed after it has been written to, the last logical block of data is reduced to the actual number of subblocks required, thus creating a fragmented block.

In order to write to a file system, free full blocks of disk space are required. Due to fragmentation, it is entirely possible to have the situation where the file system is not full, but an insufficient number of free full blocks are available to write to the file system. Replication can also cause the copy of the fragment to be distributed among disks in different failure groups. The **`mmdefragfs`** command can be used to query the current fragmented state of the file system and reduce the fragmentation of the file system.

In order to reduce the fragmentation of a file system, the **`mmdefragfs`** command migrates fragments to free space in another fragmented disk block of sufficient space, thus creating a free full block. There is no requirement to have a free full block in order to run the **`mmdefragfs`** command. The execution time of the **`mmdefragfs`** command depends on the size and allocation pattern of the file system. For a file system with a large number of disks, the **`mmdefragfs`** command will run through several iterations of its algorithm, each iteration compressing a different set of disks. Execution time is also dependent on how fragmented the file system is. The less fragmented a file system, the shorter time for the **`mmdefragfs`** command to execute.

The fragmentation of a file system can be reduced on all disks which are not suspended or stopped. If a disk is suspended or stopped, the state of the disk, not the utilization information, will be displayed as output for the **`mmdefragfs`** command.

The **`mmdefragfs`** command can be run on both a mounted or an unmounted file system, but achieves best results on an unmounted file system. Running the command on a mounted file system can cause conflicting allocation information and consequent retries to find a new free subblock of the correct size to store the fragment in.

Querying file system fragmentation

To query the status of the amount of fragmentation for a file system, specify the file system name along with the **-i** option on the **`mmdefragfs`** command.

For example, to display the current fragmentation information for file system *fs0*, enter:

```
mmdefragfs fs0 -i
```

The system displays information similar to:

```

"fs0"    10304 inodes:    457 allocated / 9847 free

                                free subblk    free
disk        disk size  in full  subblk in    %    %
name       in nSubblk  blocks   fragments free blk blk util
-----
gpfs68nsd    4390912   4270112     551  97.249  99.544
gpfs69nsd    4390912   4271360     490  97.277  99.590
(total)      8781824   8541472     1041                  99.567

```

For complete usage information, see **`mmdefragfs` command** in *IBM Storage Scale: Command and Programming Reference Guide*.

Related tasks

[Reducing file system fragmentation](#)

You can reduce the amount of fragmentation for a file system by issuing the **mmdefragfs** command, with or without a desired block usage goal.

Reducing file system fragmentation

You can reduce the amount of fragmentation for a file system by issuing the **mmdefragfs** command, with or without a desired block usage goal.

For example, to reduce the amount of fragmentation for file system **fs1** with a goal of 100% utilization, enter:

```
mmdefragfs fs1 -u 100
```

The system displays information similar to:

```
Defragmenting file system 'fs1'...

Defragmenting until full block utilization is 98.00%, currently 97.07%
27.35 % complete on Tue May 26 14:25:42 2009 ( 617882 inodes 4749 MB)
82.65 % complete on Tue May 26 14:26:02 2009 ( 1867101 inodes 10499 MB)
89.56 % complete on Tue May 26 14:26:23 2009 ( 2023206 inodes 14296 MB)
90.01 % complete on Tue May 26 14:26:43 2009 ( 2033337 inodes 17309 MB)
90.28 % complete on Tue May 26 14:27:03 2009 ( 2039551 inodes 19779 MB)
91.17 % complete on Tue May 26 14:27:23 2009 ( 2059629 inodes 23480 MB)
91.67 % complete on Tue May 26 14:27:43 2009 ( 2070865 inodes 26760 MB)
92.51 % complete on Tue May 26 14:28:03 2009 ( 2089804 inodes 29769 MB)
93.12 % complete on Tue May 26 14:28:23 2009 ( 2103697 inodes 32649 MB)
93.39 % complete on Tue May 26 14:28:43 2009 ( 2109629 inodes 34934 MB)
95.47 % complete on Tue May 26 14:29:04 2009 ( 2156805 inodes 36576 MB)
95.66 % complete on Tue May 26 14:29:24 2009 ( 2160915 inodes 38705 MB)
95.84 % complete on Tue May 26 14:29:44 2009 ( 2165146 inodes 40248 MB)
96.58 % complete on Tue May 26 14:30:04 2009 ( 2181719 inodes 41733 MB)
96.77 % complete on Tue May 26 14:30:24 2009 ( 2186053 inodes 43022 MB)
96.99 % complete on Tue May 26 14:30:44 2009 ( 2190955 inodes 43051 MB)
97.20 % complete on Tue May 26 14:31:04 2009 ( 2195726 inodes 43077 MB)
97.40 % complete on Tue May 26 14:31:24 2009 ( 2200378 inodes 43109 MB)
97.62 % complete on Tue May 26 14:31:44 2009 ( 2205201 inodes 43295 MB)
97.83 % complete on Tue May 26 14:32:05 2009 ( 2210003 inodes 43329 MB)
97.85 % complete on Tue May 26 14:32:25 2009 ( 2214741 inodes 43528 MB)
97.86 % complete on Tue May 26 14:32:55 2009 ( 2221888 inodes 43798 MB)
97.87 % complete on Tue May 26 14:33:35 2009 ( 2231453 inodes 44264 MB)
97.88 % complete on Tue May 26 14:34:26 2009 ( 2243181 inodes 45288 MB)
100.00 % complete on Tue May 26 14:35:10 2009

          free subblk          free
disk      in full subblk in % % 
name      blocks blk   fragments free blk   blk util
       before after freed   before after   before after before after
-----
nsd32    277504 287840 323   12931   2183 84.69 87.84 96.05 99.33
nsd33    315232 315456 7     580     185 96.20 96.27 99.82 99.94
nsd21    301824 303616 56    2481   666 92.11 92.66 99.24 99.80
nsd34    275904 285920 313   13598   3159 84.20 87.26 95.85 99.04
nsd30    275840 285856 313   13348   2923 84.18 87.24 95.93 99.11
nsd19    278592 288832 320   12273   1874 85.02 88.14 96.25 99.43
nsd31    276224 284608 262   12012   3146 84.30 86.86 96.33 99.04
-----
 (total) 2001120 2052128 1594  67223  14136                   97.07 99.38

Defragmentation complete, full block utilization is 99.04%.
```

See the **mmdefragfs command** in *IBM Storage Scale: Command and Programming Reference Guide* for complete usage information.

Related tasks

[Querying file system fragmentation](#)

To query the status of the amount of fragmentation for a file system, specify the file system name along with the `-i` option on the **mmdefragfs** command.

Protecting data in a file system using backup

GPFS provides ways to back up the file system user data and the overall file system configuration information.

You can use the **mmbackup** command to back up the files of a GPFS file system or the files of an independent fileset to an IBM Storage Protect server.

Alternatively, you can utilize the GPFS policy engine (**mmaplypolicy** command) to generate lists of files to be backed up and provide them as input to some other external storage manager.

The file system configuration information can be backed up using the **mmbackupconfig** command.

Note: Windows nodes do not support the **mmbackup**, **mmaplypolicy**, and **mmbackupconfig** commands.

Protecting data in a file system using the **mmbackup** command

The **mmbackup** command can be used to back up some or all of the files of a GPFS file system to IBM Storage Protect servers using the IBM Storage Protect Backup-Archive client. After files have been backed up, you can restore them using the interfaces provided by IBM Storage Protect.

The **mmbackup** command utilizes all the scalable, parallel processing capabilities of the **mmaplypolicy** command to scan the file system, evaluate the metadata of all the objects in the file system, and determine which files need to be sent to backup in IBM Storage Protect, and the deleted files that should be expired from IBM Storage Protect. Both backup and expiration take place when running **mmbackup** in the incremental backup mode.

The **mmbackup** command can interoperate with regular IBM Storage Protect commands for backup and expire operations. However, if after using **mmbackup** any IBM Storage Protect incremental or selective backup or expire commands are used, **mmbackup** needs to be informed of these activities. Use either the `-q` option or the `--rebuild` option in the next **mmbackup** command invocation to enable **mmbackup** to rebuild its shadow databases. (See **mmbackup Examples** in *IBM Storage Scale: Command and Programming Reference Guide*.)

These databases *shadow* the inventory of objects in IBM Storage Protect so that only new changes will be backed up in the next incremental **mmbackup**. Failing to do so will needlessly back up some files additional times. The shadow database can also become out of date if **mmbackup** fails due to certain IBM Storage Protect server problems that prevent **mmbackup** from properly updating its shadow database after a backup. In these cases, it is also required to issue the next **mmbackup** command with either the `-q` option or the `--rebuild` options.

The **mmbackup** command provides:

- A full backup of all files in the specified scope.
- An incremental backup of only those files that have changed or been deleted since the last backup. Files that have changed since the last backup are updated and files that have been deleted since the last backup are expired from the IBM Storage Protect server.
- Utilization of a fast scan technology for improved performance.
- The ability to perform the backup operation on a number of nodes in parallel.
- Multiple tuning parameters to allow more control over each backup.
- The ability to back up the read/write version of the file system or specific global snapshots.
- Storage of the files in the backup server under their GPFS root directory path independent of whether backing up from a global snapshot or the live file system.
- Handling of unlinked filesets to avoid inadvertent expiration of files.

Note: Avoid unlinking a fileset while running `mmbackup`. If a fileset is unlinked before `mmbackup` starts, it is handled. However, unlinking a fileset during the job could result in a failure to back up changed files as well as expiration of already backed up files from the unlinked fileset.

The `mmbackup` command supports backing up GPFS file system data to multiple IBM Storage Protect servers. The ability to partition file backups across multiple IBM Storage Protect servers is particularly useful for installations that have a large number of files. For information on setting up multiple IBM Storage Protect servers, see “[IBM Storage Protect requirements](#)” on page 245.

Unless otherwise specified, the `mmbackup` command backs up the current active version of the GPFS file system. If you want to create a backup of files at a specific point in time, first use the `mmcrsnapshot` command to create either a global snapshot or a fileset-level snapshot, and then specify that snapshot name for the `mmbackup -S` option. A global snapshot can be specified for either `--scope filesystem` or `--scope inodespace`. A fileset-level snapshot can only be specified with `--scope inodespace`.

If an unlinked fileset is detected, the `mmbackup` processing will issue an error message and exit. You can force the backup operation to proceed by specifying the `mmbackup -f` option. In this case, files that belong to unlinked filesets are not be backed up, but are removed from the expire list.

Related concepts

[Backing up file system configuration information](#)

The `mmbackupconfig` command can be used to back up vital file system configuration information. This information can later be used to restore the layout and major characteristics of the file system.

Related tasks

[Backing up a file system using the GPFS policy engine](#)

If IBM Storage Protect is not available, you can use the fast scan capabilities of the GPFS policy engine to generate lists of files to be backed up and provide them as input to some other external storage manager.

[Using APIs to develop backup applications](#)

You can develop backup applications using APIs.

Protecting data in a fileset using the `mmbackup` command

The `mmbackup` command can be used to back up an independent fileset to the IBM Storage Protect servers by using the IBM Storage Protect Backup-Archive client. After a fileset is backed up, you can restore files by using the interfaces that are provided by IBM Storage Protect.

When backing up an independent fileset, the `mmbackup` command backs up the current active version of the fileset. The path to the independent fileset root is specified with the *Directory* parameter of the `mmbackup` command.

If you want to create a backup of a fileset at a specific point in time, first use the `mmcrsnapshot` command to create a fileset-level snapshot. Next, specify that snapshot name for the `mmbackup -S` option along with the `--scope inodespace` option.

Note: The *SnapshotName* that is specified must be unique to the file system.

We recommend customers to run independent fileset backup with a unique IBM Storage Protect node name per fileset for better performance especially when multiple fileset backups are simultaneously run. This option increases the scalability and maintainability in the backup backend.

IBM Storage Protect requirements

The `mmbackup` command requires an IBM Storage Protect client and server environment to perform a backup operation.

For details on the supported versions of IBM Storage Protect, client and server installation and setup, and include and exclude lists, see the [IBM Tivoli® Storage Manager V7.1.7 documentation](#).

1. Ensure that the supported versions of the IBM Storage Protect client and server are installed. See the [IBM Storage Scale FAQ in IBM Documentation](#).
2. Ensure that the IBM Storage Protect server and clients are configured properly for backup operations.

3. If you are using multiple IBM Storage Protect servers to protect data, ensure that the IBM Storage Protect servers are set up properly.
4. Ensure the required `dsm.sys` and `dsm.opt` configuration files are present in the IBM Storage Protect configuration directory on each node used to run `mmbbackup` or named in a node specification with `-N`.
5. If you want to include or exclude specific files or directories by using include-exclude lists, ensure that the lists are set up correctly before you invoke the `mmbbackup` command.

The `mmbbackup` command uses an IBM Storage Protect include-exclude list for including and excluding specific files or directories. See the Tivoli documentation for information about defining an include-exclude list.

Note: IBM Storage Protect interprets its include and exclude statements in a unique manner that is not precisely matched by the GPFS `mmapplypolicy` file selection language. The essential meaning of each supported include or exclude statement is followed, but the commonly used IBM Storage Protect idiom of excluding everything as the last statement and including selective directory or file name patterns in prior statements should not be used with GPFS and `mmbbackup`. The exclusion pattern of `"//*"` is interpreted by `mmapplypolicy` to exclude everything, and no data is backed up.

A very large include-exclude list can decrease backup performance. Use wildcards and eliminate unnecessary include statements to keep the list as short as possible.

6. If more than one node is used to perform the backup operation (`mmbbackup -N` option):
 - The `mmbbackup` command verifies that the IBM Storage Protect Backup-Archive client versions and configuration are correct before executing the backup. Any nodes that are not configured correctly will be removed from the backup operation. Ensure that IBM Storage Protect clients are installed and at the same version on all nodes that will invoke the `mmbbackup` command or participate in parallel backup operations.
 - Ensure that IBM Storage Protect is aware that the various IBM Storage Protect clients are all working on the same file system, not different file systems having the same name on different client machines. This is accomplished by using proxy nodes for multiple nodes in the cluster. See the IBM Storage Protect documentation for recommended settings for GPFS cluster nodes setup.
7. Restoration of backed-up data must be done using IBM Storage Protect interfaces. This can be done with the client command-line interface or the IBM Storage Protect web client. The IBM Storage Protect web client interface must be made operational if you wish to use this interface for restoring data to the file system from the IBM Storage Protect server.
8. When more than one IBM Storage Protect server is referenced in the `dsm.sys` file, `mmbbackup` uses all listed IBM Storage Protect servers by default. To use only a select IBM Storage Protect server or the servers that are listed in `dsm.sys`, use the `mmbbackup --tsm-servers` option. When more than one IBM Storage Protect server is used for backup, the list and the order specified should remain constant. If additional IBM Storage Protect servers are added to the backup later, add them to the end of the list that is specified with the `mmbbackup --tsm-servers` option.
9. IBM Storage Protect does not support special characters in the path names and in some cases cannot back up a path name that has special characters. A limited number of special characters are supported on IBM Storage Protect client 6.4.0.0 and later versions with client options `WILDCARDSARELITERAL` and `QUOTESARELITERAL`. Use these IBM Storage Protect options with the `mmbbackup --noquote` option if you have path names with special characters. The `mmbbackup` command does not back up path names containing any newline, `Ctrl+X`, or `Ctrl+Y` characters. If the `mmbbackup` command finds unsupported characters in the path name, it writes that path to a file called `mmbbackup.unsupported.tsmserver` at the root of the `mmbbackup` record directory (by default it is the root of the file system).



Attention: If you are using the IBM Storage Protect Backup-Archive client command line or web interface to do back up, use caution when you unlink filesets that contain data backed up by IBM Storage Protect. IBM Storage Protect tracks files by path name and does not track filesets. As a result, when you unlink a fileset, it appears to IBM Storage Protect that you deleted the contents of the fileset. Therefore, the IBM Storage Protect Backup-Archive client inactivates the data on the IBM Storage Protect server, which may result in the loss of backup data during the expiration process.

Migrating to **mmbackup** from IBM Storage Protect-interface-based backup

File systems that are backed up using the IBM Storage Protect interface can be converted to use the **mmbackup** command to take advantage of the performance offered by **mmbackup** fast scan technology.

A full backup is not required or necessary when moving from backup using the IBM Storage Protect interface to the **mmbackup** command.

The **mmbackup** command uses one or more shadow database files to determine changes in the file system. To convert from the IBM Storage Protect interface backup to **mmbackup**, one must create the shadow database file or files by using the **--rebuild** option of **mmbackup**. The rebuild option queries the existing IBM Storage Protect server or servers and creates a shadow database of the files currently backed up in IBM Storage Protect. After the shadow database file or files are generated, **mmbackup** can be used for all future incremental or full backups.

Note: If using multiple IBM Storage Protect servers to back up a file system, use the **mmbackup --tsm-servers** option to ensure that the proper servers participate in the backup job.

Tuning backups with the **mmbackup** command

You can tune backups with the **mmbackup** command.

The **mmbackup** command performs all its work in three major steps, and all of these steps potentially use multiple nodes and threads:

1. The file system is scanned with **mmapplypolicy**, and a list is created of every file that qualifies and should be in backup for each IBM Storage Protect server in use. The existing shadow database and the list generated are then compared and the differences between them yield:
 - Objects deleted recently that should be marked inactive (expire)
 - Objects modified or newly created to back up (selective)
 - Objects modified without data changes; owner, group, mode, and migration state changes to update (incremental)
2. Using the lists created in step “1” on page 247, **mmapplypolicy** is run for files that should be marked inactive (expire).
3. Using the lists created in step “1” on page 247, **mmapplypolicy** is run for selective or incremental backup.

The **mmbackup** command has several parameters that can be used to tune backup jobs. During the scanning phase, the resources **mmbackup** will utilize on each node specified with the **-N** parameter can be controlled:

- The **-a IscanThreads** parameter allows specification of the number of threads and sort pipelines each node will run during the parallel inode scan and policy evaluation. This parameter affects the execution of the high-performance protocol that is used when both the **-g** and **-N** parameters are specified. The default value is 2. Using a moderately larger number can significantly improve performance, but might strain the resources of the node. In some environments a large value for this parameter can lead to a command failure.

Tip: Set this parameter to the number of CPU cores implemented on a typical node in your GPFS cluster.

- The **-n DirThreadLevel** parameter allows specification of the number of threads that will be created and dispatched within each **mmapplypolicy** process during the directory scan phase.

During the execution phase for expire, **mmbackup** processing can be adjusted as follows:

- Automatic computation of the ideal expire bunch count. The number of objects named in each file list can be determined, separately from the number in a backup list, and automatically computed, if not specified by the user.
- As an alternative to the automatic computation, the user can control expire processing as follows:
 - The **--max-expire-count** parameter can be used to specify a bunch-count limit for each **dsmc expire** command. This parameter cannot be used in conjunction with **-B**.

- The `--expire-threads` parameter can be used to control how many threads run on each node running `dsmc expire`. This parameter cannot be used in conjunction with `-m`.

During the execution phase for backup, `mmbackup` processing can be adjusted as follows:

- Automatic computation of ideal backup bunch count. The number of objects named in each file list can be determined, separately from the number in an expire list, and automatically computed, if not specified by the user.
- As an alternative to the automatic computation, the user can control backup processing as follows:
 - The `--max-backup-count` parameter can be used to specify a bunch-count limit for each `dsmc selective` or `dsmc incremental` command. This parameter cannot be used in conjunction with `-B`.
 - The `--backup-threads` parameter can be used to control how many threads run on each node running backup. This parameter cannot be used in conjunction with `-m`.
 - The `--max-backup-size` parameter can be used to further limit the size of a backup bunch by the overall size of all files listed in any single bunch list.
 - The `--max-incremental-backup-count` parameter can be used to limit the maximum number of objects in any `dsmc incremental` backup command to be `n` objects in a bunch. This option cannot be used with the `-B`, `--max-backup-count`, or `-t full`.
 - The `--max-selective-backup-count` parameter can be used to limit the maximum number of objects in any single `dsmc selective` backup command to be `n` objects in a bunch. This option cannot be used with the `-B` or `--max-backup-count` options.

For more information on the `mmbackup` tuning parameters, see **`mmbackup command`** in *IBM Storage Scale: Command and Programming Reference Guide*.

MMBACKUP_PROGRESS_CALLOUT environment variable

The `MMBACKUP_PROGRESS_CALLOUT` environment variable specifies the path to a program or script to be called during `mmbackup` execution with a formatted argument.

The `$progressCallOut` function is executed if the path `$progressCallOut` names a valid, executable file and one of the following is true:

- The message class provided with this message is 0.
Or
- At least `$progressInterval` seconds has elapsed.
Or
- The `$progressContent` mask has a bit set which matches a bit set in the message class provided with this message.

The `$progressCallOut` function is executed during `mmbackup` with a single argument consisting of the following colon-separated values:

```
"$JOB:$FS:$SERVER:$NODENAME:$PHASE:$BCKFILES:$CHGFILES:$EXPFILES:\`$FILESBACKEDUP:$FILESEXPIRED:$ERRORS:$TIME:$TOTALSIZE:$SIZEBACKEDUP"
```

Where:

JOB

Specifies the literal backup string to identify this component.

FS

Specifies the file system device name.

SERVER

Specifies the IBM Storage Protect server currently used for backup.

NODENAME

Specifies the name of the node where `mmbackup` was started.

PHASE

Specifies either synchronizing, scanning, selecting files, expiring, backing up, analyzing, or finishing.

BCKFILES

Specifies the total number of files already backed up, or stored, on the IBM Storage Protect server. Starts as the count of all normal mode records in all the current shadow databases in use. If QUERY is being executed, it will start as the count of files found on the IBM Storage Protect server. It will stay constant until the backup job is complete.

CHGFILES

Specifies the number of changed files. This value starts as 0 and changes to the total number of changed files destined for the current server, and then stays at that value.

EXPFILES

Specifies the number of expired files. This value starts as 0 and changes to the total number of files marked for expiration at the current server, and then stays at that value.

FILESBACKEDUP

Specifies the number of files that were backed up during this backup job. This value remains 0 until phase backing up is reached, and then it increases until dsmc finishes. This value increases while dsmc selective jobs are running and is calculated by IBM Storage Protect output. If the backup job fails before completion, some output may indicate files backed up but not counted. This value always increases.

FILESEXPIRED

Specifies the number of files that expired during this expire job. This value remains 0 until phase expiring is reached, and then it increases until dsmc finishes. This value increases while dsmc expire jobs are running and is calculated by IBM Storage Protect output. If the backup job fails before completion, some output may indicate files expired but not counted. This value always increases.

ERRORS

Specifies the number of errors, not warnings or informational messages, which occurred during processing.

TIME

Specifies the time stamp as a ctime or number of seconds since the Epoch.

\$TOTALSIZE

Specifies the data in bytes to be backed up. This information is available only when MMBACKUP_PROGRESS_CONTENT environment variable contains 0x08.

\$SIZEBACKEDUP

Specifies the data in bytes that have been backed up so far. This information is available only when MMBACKUP_PROGRESS_CONTENT environment variable contains 0x08.

Backing up a file system using the GPFS policy engine

If IBM Storage Protect is not available, you can use the fast scan capabilities of the GPFS policy engine to generate lists of files to be backed up and provide them as input to some other external storage manager.

This process typically includes:

- Creating a policy file with LIST rules and associated criteria to generate the desired lists
- Optionally, creating a snapshot to obtain a consistent copy of the file system at a given point in time
- Running the **mmapplypolicy** command to generate the lists of files to back up
- Invoking the external storage manager to perform the actual backup operation

For more information on GPFS policies and rules refer to Chapter 39, “Information lifecycle management for IBM Storage Scale,” on page 525.

Related concepts

[Backing up file system configuration information](#)

The **mmbackupconfig** command can be used to back up vital file system configuration information. This information can later be used to restore the layout and major characteristics of the file system.

Related tasks

[Protecting data in a file system using the mmbackup command](#)

The mmbackup command can be used to back up some or all of the files of a GPFS file system to IBM Storage Protect servers using the IBM Storage Protect Backup-Archive client. After files have been backed up, you can restore them using the interfaces provided by IBM Storage Protect.

[Using APIs to develop backup applications](#)

You can develop backup applications using APIs.

Backing up file system configuration information

The **mmbackupconfig** command can be used to back up vital file system configuration information. This information can later be used to restore the layout and major characteristics of the file system.

The **mmbackupconfig** command creates a file that includes:

- Disk information (NSD names, sizes, failure groups)
- Storage pool layout
- Filesets and junction points
- Policy file rules
- Quota settings and current limits
- File system parameters (block size, replication factors, number of inodes, default mount point, and so on)

The output file generated by the **mmbackupconfig** command is used as input to the **mmrestoreconfig** command.

Note: The **mmbackupconfig** command only backs up the file system configuration information. It does not back up any user data or individual file attributes.

It is recommended that you store the output file generated by **mmbackupconfig** in a safe location.

Related tasks

[Protecting data in a file system using the mmbackup command](#)

The mmbackup command can be used to back up some or all of the files of a GPFS file system to IBM Storage Protect servers using the IBM Storage Protect Backup-Archive client. After files have been backed up, you can restore them using the interfaces provided by IBM Storage Protect.

[Backing up a file system using the GPFS policy engine](#)

If IBM Storage Protect is not available, you can use the fast scan capabilities of the GPFS policy engine to generate lists of files to be backed up and provide them as input to some other external storage manager.

[Using APIs to develop backup applications](#)

You can develop backup applications using APIs.

Using APIs to develop backup applications

You can develop backup applications using APIs.

IBM has supplied a set of subroutines that are useful to create backups or collect information about all files in a file system. Each subroutine is described in *Programming interfaces in IBM Storage Scale: Command and Programming Reference Guide*. These subroutines are more efficient for traversing a file system, and provide more features than the standard POSIX interfaces. These subroutines operate on a global snapshot or on the active file system. They have the ability to return all files, or only files that have changed since some earlier snapshot, which is useful for incremental backup.

A typical use of these subroutines is the following scenario:

1. Create a global snapshot using the **mmcrlsnapshot** command. For more information on snapshots, see the *IBM Storage Scale: Command and Programming Reference Guide*.

2. Open an inode scan on the global snapshot using the **gpfs_open_inodescan()** or **gpfs_open_inodescan64()** subroutine.
3. Retrieve inodes using the **gpfs_next_inode()** or **gpfs_next_inode64()** subroutine.
4. Read the file data:
 - a. Open the file using the **gpfs_iopen()** or **gpfs_iopen64()** subroutine.
 - b. Read the file using the **gpfs_iread()**, **gpfs_ireadx()**, **gpfs_ireaddir()**, or **gpfs_ireaddir64()** subroutines.
 - c. Close the file using the **gpfs_iclose()** subroutine.

The **gpfs_ireadx()** subroutine is more efficient than **read()** or **gpfs_iread()** for sparse files and for incremental backups. The **gpfs_ireaddir()** or **gpfs_ireaddir64()** subroutine is more efficient than **readdir()**, because it returns file type information. There are also subroutines for reading symbolic links, **gpfs_ireadlink()** or **gpfs_ireadlink64()** and for accessing file attributes, **gpfs_igetattr()**.

Related concepts

[Backing up file system configuration information](#)

The **mmbackupconfig** command can be used to back up vital file system configuration information. This information can later be used to restore the layout and major characteristics of the file system.

Related tasks

[Protecting data in a file system using the mmbackup command](#)

The **mmbackup** command can be used to back up some or all of the files of a GPFS file system to IBM Storage Protect servers using the IBM Storage Protect Backup-Archive client. After files have been backed up, you can restore them using the interfaces provided by IBM Storage Protect.

[Backing up a file system using the GPFS policy engine](#)

If IBM Storage Protect is not available, you can use the fast scan capabilities of the GPFS policy engine to generate lists of files to be backed up and provide them as input to some other external storage manager.

Scale Out Backup and Restore (SOBAR)

Scale Out Backup and Restore (SOBAR) is a specialized mechanism for data protection against disaster only for GPFS file systems that are managed by IBM Storage Protect for Space Management.

For such systems, the opportunity exists to:

1. Pre-migrate all file data into the IBM Storage Protect for Space Management storage
2. Take a snapshot of the file system structural metadata
3. Save a backup image of the file system structure

This metadata image backup, consisting of several image files, can be safely stored in the backup pool of the IBM Storage Protect server and later used to restore the file system in the event of a disaster.

The SOBAR utilities include the **mmbackupconfig**, **mmrestoreconfig**, **mmimgbackup**, and **mmimgrestore** commands. The **mmbackupconfig** command will record all the configuration information about the file system to be protected and the **mmimgbackup** command performs a backup of GPFS file system metadata. The resulting configuration data file and the metadata image files can then be copied to the IBM Storage Protect server for protection.

In the event of a disaster, the file system can be recovered by recreating the necessary NSD disks, restoring the file system configuration with the **mmrestoreconfig** command, and then restoring the image of the file system with the **mmimgrestore** command. The **mmrestoreconfig** command must be run prior to running the **mmimgrestore** command. SOBAR will reduce the time needed for a complete restore by utilizing all available bandwidth and all available nodes in the GPFS cluster to process the image data in a highly parallel fashion. It will also permit users to access the file system before all file data has been restored, thereby minimizing the file system down time. Recall from IBM Storage Protect for Space Management of needed file data is performed automatically when a file is first accessed.

These commands cannot be run from a Windows node.

For the full details of the SOBAR procedures and requirements, see *Scale Out Backup and Restore (SOBAR) in IBM Storage Scale: Command and Programming Reference Guide*.

Scheduling backups using IBM Storage Protect scheduler

The IBM Storage Protect scheduler typically utilizes the IBM Storage Protect Backup-Archive client backup commands that should be avoided in the IBM Storage Scale setup. Instead, you can configure the IBM Storage Protect client schedule to call a script as described in the following steps.

For scheduled events to occur on the client, you must configure the client scheduler to communicate with the IBM Storage Protect server. This is in addition to the following steps. For example, you might need to start the dsmcad service or add MANAGEDSERVICES schedule to the corresponding IBM Storage Protect stanza in dsm.sys on the client node. For more information, see *Configuring the scheduler* in the IBM Storage Protect documentation.

For the following steps, these example values are assumed:

```
client-node-proxyname (asnodenname)      => proxy-cluster1
Node to be used for the schedule (aka nodename) => gpfs-node1
tsm server name                           => tsm1
file system to be backed up              => gpfs0
global snapshot name (created for backup job) => BKUPsnap
schedule name on the TSM server          => proxy-cluster1_sched
```

1. On the IBM Storage Protect server, define the schedule using the following command:

```
define schedule standard proxy-cluster1_sched type=client action=command
objects=/usr/bin/my-mmbackup-script.sh starttime=05:00:00 startdate=today
```

2. On the IBM Storage Protect server, associate the schedule with the IBM Storage Scale proxy node using the following command:

```
define association standard proxy-cluster1_sched proxy-cluster1
```

3. Create the backup script on the IBM Storage Scale node.

Note: The following example script must be extended to log the output into files so that verification or troubleshooting can be done afterward. Additional options such as --noquote might be needed depending on the specific needs of the environment.

```
#!/bin/bash
/usr/lpp/mmfs/bin/mmcrsnapshot gpfs0 BKUPsnap
/usr/lpp/mmfs/bin/mmbackup gpfs0 -t incremental --tsm-servers tsm1
/usr/lpp/mmfs/bin/mmdelsnapshot gpfs0 BKUPsnap
```

4. On one of the IBM Storage Protect client nodes, verify the schedule using the following command:

```
dsmc q sched
```

Configuration reference for using IBM Storage Protect with IBM Storage Scale

When using the IBM Storage Protect client in an IBM Storage Scale environment, several options in the dsm.sys and dsm.opt configuration files need to be taken into consideration.

Note: Refer to the latest IBM Storage Protect documentation for the latest information on the mentioned settings.

Options in the IBM Storage Protect configuration file dsm.sys

This topic describes the options in the IBM Storage Protect configuration file dsm.sys.

Important: While the IBM Storage Protect client configuration file dsm.sys can contain node-specific information, it cannot be copied from node to node without touching or correcting the corresponding node-specific information.

Exclude options

You can exclude directories and files from backup either in IBM Storage Protect or in the dsm.sys file.

The **mmbbackup** command excludes the following folders from the scan by default:

- .mmbbackup* - folder in location that is specified by **MMBACKUP_RECORD_ROOT** such as /ibm/gpfs0/.mmbbackupCfg
- .mmLockDir - folder in the root of the file system
- .SpaceMan - folder anywhere in the file system
- .TsmCacheDir - folder anywhere in the file system

Do not create exclude statements for snapshots. Snapshots are specially handled automatically by **mmbbackup** and IBM Storage Protect options.

Note: Defining many exclude rules can negatively impact the performance of backup.

Paths that contain blank space characters

The **mmbbackup** command does not interpret blanks spaces in path names correctly in include and exclude rules in the dsm.sys file. If a rule contains a path with blank spaces, IBM Storage Protect can reject the resulting query from the **mmbbackup** command or can include or exclude files that you did not want to include or exclude.

To work around this limitation, you can specify a token delimiter string other than a blank space for the output from the `dsmc query inclexcl` command. For example, to set two colons (::) as the token delimiter string, add the following command to the dsm.opt file:

```
TESTFLAGS PARSEABLE_INCLEXCL "::"
```

You can now create an exclude rule for a directory path that contains blank spaces, as in the following example:

```
EXCLUDE.DIR "/gpfs1/new tool scripts/bin"
```

The command `dsmc q inclexcl` displays the following information for the exclude rule:

```
Excl:::Directory:::/gpfs1/new tool scripts/bin
```

Include options

IBM Storage Protect provides include options and exclude options and can correctly process a set of includes and excludes. However, the **mmbbackup** command is apt to misinterpret a mixture of include rules and exclude rules, especially when there are overlapping pattern sequences. Therefore, it is a good practice to avoid usage of the include rules with the **mmbbackup** command. For an exception, see the next topic.

Server management class assignments

Special consideration is needed when IBM Storage Protect server management class definitions are used. The corresponding include statements must be applied to any dsm.sys and not applied on the IBM Storage Protect server.

IBM Storage Protect users might be familiar with dynamic management class assignments available when using IBM Storage Protect **dsmc** commands to backup files. This is not the case with **mmbackup**. Only objects identified by **mmbackup** as requiring a backup gets the needed management class update that results when the administrator alters the management class assignment in the **dsm.sys** file. Therefore, only by running a complete backup of all affected objects can a management class update be guaranteed.

Despite the recommendation, to never use the include statements in **dsm.sys** when the IBM Storage Protect management class designation is needed, the use of an include statement with the management class specification is required. In these cases, do the following steps:

1. In the IBM Storage Protect client configuration file **dsm.sys**, arrange the include and exclude statements as follows:

- a. Place all the include statement first in the file along with the management class definitions.
- b. Add the exclude statements under the include statements.
- c. Ignore the ordering precedence rules that are defined in the IBM Storage Protect documentation that is regarding the ordering of these statements. Management class include statements must be listed before the exclude statements to work properly with **mmbackup**.

Note: Do not add include statements after exclude statements. Do not add exclude statements before include statements.

2. Before starting the **mmbackup** job, set the following environment variable:

```
export MMBACKUP_IGNORE_INCLUDE=1
```

Note:

- The include statements have no effect on the file system scan candidate selection in **mmapplypolicy** because the rules for include do not result in SQL statements that are generated with **MMBACKUP_IGNORE_INCLUDE** activated.
- The include statements do not overrule the exclude statements that can be the case sometimes with **mmapplypolicy** policy rules that are generated from include and exclude formulation in IBM Storage Protect. It is recommended to never have overlapping patterns of any type with both include and exclude statements.

Usage of an IBM Storage Protect proxy node (asnodename option)

In a cluster, an operation that needs to scale, is usually run on more than one node, for example backup activities. To use the services of an IBM Storage Protect server from any of the configured cluster backup nodes, the administrator needs to specify a proxy node. This proxy node needs to be created on the IBM Storage Protect server similar to all other cluster backup nodes that need to be registered on the IBM Storage Protect server before they can be used. On all cluster backup nodes, set the **asnodename** option for the desired proxy-client node to be used in the corresponding stanza of the **dsm.sys** configuration file.

Important IBM Storage Protect client configuration option

Option name	Remarks	Context
ASNODENAME \$client-node-proxyname	Use the proxy node name (asnodename) instead of the cluster node name (nodename) to process cluster operations independent of a node name that is required for restore processing.	General

Related concepts

[Options in the IBM Storage Protect configuration file **dsm.opt**](#)

This topic describes the options in the IBM Storage Protect configuration file `dsm.opt`.

[Base IBM Storage Protect client configuration files for IBM Storage Scale usage](#)

This topic lists all the Base IBM Storage Protect client configuration files and their examples for IBM Storage Scale.

Options in the IBM Storage Protect configuration file `dsm.opt`

This topic describes the options in the IBM Storage Protect configuration file `dsm.opt`.

Note: Use the `DSM_CONFIG` environment variable to point to a specific `dsm.opt` file.

Special character handling

For IBM Storage Scale file systems with special characters frequently used in the names of files or directories, backup failures might occur. Known special characters that require special handling include: *, ?, ", ', carriage return, and the new line character.

In such cases, enable the IBM Storage Protect client options `WILDCARDSARELITERAL` and `QUOTESARELITERAL` on all nodes that are used in backup activities and make sure that the `mmbackup` option `--noquote` is used when invoking `mmbackup`.

Note: The characters control-X and control-Y are not supported by IBM Storage Protect. Therefore, the use of these characters in file names in IBM Storage Scale file systems results in these files not getting backed up to IBM Storage Protect.

Important IBM Storage Protect client configuration options

Option name	Remarks	Context
<code>QUOTESARELITERAL</code> [YES NO]	Requires the use of <code>mmbackup</code> with option <code>--noquote</code> if this is set to YES.	General
<code>WILDCARDSARELITERAL</code> [YES NO]	To handle the wildcard characters * and ? in file and folder names.	General
<code>HSMDISABLEAUTOMIGDAEMONS</code> [YES NO]	To prevent the IBM Storage Protect for Space Management automigration daemons from starting. Instead, the <code>mmapplypolicy</code> scan engine is used to identify migration candidates.	IBM Storage Protect for Space Management

Option name	Remarks	Context
SKIPACLUPDATECHECK [YES NO]	<p>Requires UPDATETIME to be enabled if this is set to YES.</p> <p>Using the SKIPACLUPDATECHECK option also omits checking for changes in the extended attributes (EAs) on Linux and AIX systems. Using this setting ensures that a file only gets backed up when the content of the file changes, not when only the ACL or EAs change. The backup of file after content changes then also includes the current ACL or EAs of the file.</p>	General
SKIPACL [YES NO]	<p>Requires UPDATETIME to be enabled if this is set to YES.</p> <p>Using the skipacl option also omits EAs on Linux and AIX systems. Using this option can be considered when static ACL structures are used that can be reestablished through another tool or operation external to the IBM Storage Protect restore operation. If you are using this approach, ensure that the ACL is restored or established by inheritance, to avoid an unauthorized access to a recently restored file or directory.</p> <p>After enabling this option, the ACL or EA is no longer backed up.</p>	General
UPDATETIME [YES NO]	<p>This is to check the change time (ctime) attribute during a backup or archive operation. It is required to perform operations such as determining ACL changes.</p>	General

Related concepts

[Options in the IBM Storage Protect configuration file dsm.sys](#)

This topic describes the options in the IBM Storage Protect configuration file dsm.sys.

[Base IBM Storage Protect client configuration files for IBM Storage Scale usage](#)

This topic lists all the Base IBM Storage Protect client configuration files and their examples for IBM Storage Scale.

Base IBM Storage Protect client configuration files for IBM Storage Scale usage

This topic lists all the Base IBM Storage Protect client configuration files and their examples for IBM Storage Scale.

Important: While the IBM Storage Protect client configuration file `dsm.sys` can contain node-specific information, it cannot be copied from node to node without touching or correcting the corresponding node-specific information.

The following are example contents of IBM Storage Protect configuration files.

Contents of `dsm.sys`

Note: Substitute the variables starting with '\$' with your own required value. See the following example values of variables.

```
Servername $servername
  COMMMethod      TCPPIP
  TCPPort         $serverport
  TCPServeraddress $serverip
*   TCPAdminPort    $serveradminport
  TCPBuffsize      512
  PASSWORDACCESS   generate
* Place your exclude rules here or configure as cloptset on TSM server
  ERRORLOGName    $errorlog
  ASNODENAME       $client-node-proxyname
  NODENAME         $localnodename
```

Example values of variables used in `dsm.sys`

```
serverport=1500
serverip=myTSMserver.mydomain.org          OR     serverip=1.2.3.4
serveradminport=1526
errorlog=/var/log/mylogs/dsmerror.log
client-node-proxyname=proxy-cluster1
localnodename=gpfs-node1
```

Contents of `dsm.opt`

```
* Special character test flags
QUOTESARELITERAL YES
WILDCARDSARELITERAL YES
* to take traces just remove the * from the next two lines:
*TRACEFLAG SERVICE
*TRACEFILE /tmp/tsmtrace.txt
```

Contents of `dsm.opt` when IBM Storage Protect for Space Management is used

```
* HSM: Write extObjID to DAPI attribute 'IBMexID' for migrated/pre-migrated files
HSMEXTOBJIDATTR yes
* HSM: Deactivate HSM Automigration and Scout search engine as this will be done by GPFS
HSMDISABLEAUTOMIGDAEMONS YES
* HSM file aggregation of small files
HSMGROUPedmigrate yes
* HSM: Determines if files that are less than 2 minutes old can be migrated during selective
migration
hsmenableimmediatemigrate yes
```

Related concepts

[Options in the IBM Storage Protect configuration file `dsm.sys`](#)

This topic describes the options in the IBM Storage Protect configuration file `dsm.sys`.

[Options in the IBM Storage Protect configuration file `dsm.opt`](#)

This topic describes the options in the IBM Storage Protect configuration file `dsm.opt`.

Restoring a subset of files or directories from a local file system snapshot

You can restore a subset of files or directories from a local snapshot of a file system in case of accidental deletion.

Ensure the following before you begin:

- You have the full path to the files or directories that you want to restore. The path must include the file system to which these files or directories belong.
- You know which snapshot contains the files or directories that you want to restore.
- You have created a restore directory to which these files or directories are to be restored to avoid accidentally overwriting files or directories.

For information on how to create and maintain snapshots, see [Chapter 40, “Creating and maintaining snapshots of file systems,” on page 601](#).

Use these steps to restore files or directories from a local file system snapshot.

1. Use the **mmlssnapshot device** command to list the snapshots in the file system and make a note of the snapshot that contains the files and directories that you want to restore.

device is the name of the file system.

```
# mmlssnapshot fs1
Snapshots in file system fs1:
Directory      SnapId Status   Created           Fileset
fileset_test1    1     Valid    Mon Mar 23 09:20:37 2015  nfs-ganesha
filesystem_test2 2     Valid    Mon Mar 23 11:12:59 2015
```

2. Use the **mmsnapdir device** command to obtain the name of the snapshot directory for the file system snapshot that you have identified.

In the following example, the fileset snapshot directory is called `.snapshots`.

```
# mmsnapdir fs1
Fileset snapshot directory for "fs1" is ".snapshots" (root directory only)
Global snapshot directory for "fs1" is ".snapshots" in root fileset
```

3. Use the **mmlsfs device -T** command to determine the default mount point of the file system.

In the following example, the default mount point is `/gpfs/fs1`.

```
# mmlsfs fs1 -T
flag  value      description
----- -----
-T    /gpfs/fs1  Default mount point
```

4. Use the full path to the files and directories that you want to restore and the default mount point that you have determined to obtain the truncated path to the files and directories.

For example:

```
Full path to the file: /gpfs/fs1/nfs-ganesha/test1/
Default mount point: /gpfs/fs1
Truncated path:       /nfs-ganesha/test1/
```

5. Change the directory to the full snapshot path of the file or the directory to verify.

The full snapshot path is:

filesystem_default_mountpoint/snapshot_directory/snapshot_name/truncated_path

The full snapshot path using examples in the preceding steps is:

/gpfs/fs1/.snapshots/filesystem_test2/nfs-ganesha/test1/

6. Do one of the following steps depending on whether you want to restore a file or a directory:

- If you want to restore a file, use the following command:

cp -p full_snapshot_path/file_name restore_directory

- If you want to restore a directory, change the directory to the *restore_directory* and use the following command:

tar -zcf tar_file_name full_snapshot_path/directory_name

Restoring a subset of files or directories from a local fileset snapshot

You can restore a subset of files or directories from a local snapshot of an independent fileset in case of accidental deletion.

Ensure the following before you begin:

- You have the full path to the files or directories that you want to restore. The path must include the file system to which these files or directories belong.
- You know which snapshot contains the files or directories that you want to restore.
- You have created a restore directory to which these files or directories are to be restored to avoid accidentally overwriting files or directories.

For information on how to create and maintain snapshots, see [Chapter 40, “Creating and maintaining snapshots of file systems,” on page 601](#).

Use these steps to restore files or directories from a local fileset snapshot.

1. Use the **mmlssnapshot device** command to list the snapshots in the file system and make a note of the snapshot that contains the files and directories that you want to restore.

device is the name of the file system.

```
# mmlssnapshot fs1
Snapshots in file system fs1:
Directory      SnapId  Status   Created           Fileset
fileset_test1    1        Valid    Mon Mar 23 09:20:37 2015  nfs-ganesha
filesystem_test2 2        Valid    Mon Mar 23 11:12:59 2015
```

2. Use the **mmsnapdir device** command to obtain the name of the snapshot directory for the fileset snapshot that you have identified.

In the following example, the fileset snapshot directory is called `.snapshots`.

```
# mmsnapdir fs1
Fileset snapshot directory for "fs1" is ".snapshots" (root directory only)
Global snapshot directory for "fs1" is ".snapshots" in root fileset
```

3. Use the **mmlsfileset device** command to verify that the fileset status is linked and to determine the full path of the fileset.

In the following example, all filesets are linked and the paths are in the third column.

```
# mmlsfileset fs1
Filesets in file system 'fs1':
Name      Status   Path
root     Linked   /gpfs/fs1
nfs-ganesha  Linked   /gpfs/fs1/nfs-ganesha
nfs-ganesha2  Linked   /gpfs/fs1/nfs-ganesha2
```

```
nfs-ganesha3  Linked    /gpfs/fs1/nfs-ganesha3  
nfs-ganesha4  Linked    /gpfs/fs1/nfs-ganesha4
```

4. Use the full path to the files and directories that you want to restore and the fileset path that you have determined to obtain the truncated path to the files and directories.

For example:

```
Full path to the file: /gpfs/fs1/nfs-ganesha/test1/  
Fileset path:          /gpfs/fs1/nfs-ganesha  
Truncated path:        /test1/
```

5. Change the directory to the full snapshot path of the file or the directory to verify.

The full snapshot path is:

fileset_path/snapshot_directory/snapshot_name/truncated_path

The full snapshot path using examples in the preceding steps is:

/gpfs/fs1/nfs-ganesha/.snapshots/fileset_test1/test1/

6. Do one of the following steps depending on whether you want to restore a file or a directory:

- If you want to restore a file, use the following command:

cp -p full_snapshot_path/file_name restore_directory

- If you want to restore a directory, change the directory to the *restore_directory* and use the following command:

tar -zcf tar_file_name full_snapshot_path/directory_name

Restoring a subset of files or directories from local snapshots using the sample script

You can restore a subset of files or directories from local snapshots using a sample script in case of accidental deletion.

- The **mmcdpsnapqueryrecover** sample script only works on the Linux operating system.
- The sample script retrieves files or directories from all file system and fileset snapshots on the system and presents a list of files that you can choose to restore.
- Regular files are simply copied into the user-specified directory. If the user specifies a directory to be retrieved, the directory is copied into the user-specified directory as a compressed tar file.
- Files and directories that contain spaces in their names can also be retrieved.

Use the **mmcdpsnapqueryrecover** sample script to restore files or directories from snapshots into the user-specified *restorePath* directory as follows.

1. Use the following command to list all copies of a file or directory in a file system or fileset snapshot.

```
/usr/lpp/mmfs/samples/ilm/mmcpsnapqueryrecover.sh Device \  
--file-path fsPath --destination-dir restorePath
```

Where:

- *device* is the name of the file system.
- *file-path* is the full file path.
- *destination-dir* is the full path of the restore directory.

For example, to get all copies of the file /gpfs0/gplssnapshot in the file system gpfs0 and with /opt as the restore directory, enter the following:

```
/usr/lpp/mmfs/samples/ilm/mmcpsnapqueryrecover.sh /dev/gpfs0 \  
--file-path /gpfs0/gplssnapshot --destination-dir /opt
```

All copies of the specified file are listed as follows:

```
Found regular file in filesystem snapshot: restorFiles1
1) 5743 Jan 9 08:34 /gpfs0/.snapshots/restorFiles1/gplssnapshot

Found regular file in filesystem snapshot: restorFiles3
2) 5882 Jan 9 08:34 /gpfs0/.snapshots/restorFiles3/gplssnapshot

Found regular file in filesystem snapshot: Restore1
3) 5886 Jan 14 12:33 /gpfs0/.snapshots/Restore1/gplssnapshot

Found regular file in filesystem snapshot: Restore2
4) 5886 Jan 14 12:33 /gpfs0/.snapshots/Restore2/gplssnapshot

Found regular file in filesystem snapshot: global1
5) 5886 Jan 14 12:33 /gpfs0/.snapshots/global1/gplssnapshot

Which copy of the file/directory (1-5) would you like to restore?
```

- From the list, select the file that you want to restore by entering the corresponding number.

For example:

```
Which copy of the file/directory (1-5) would you like to restore? 2
```

The copy number 2 is restored to the /opt directory.

Creating and managing file systems by using GUI

You can create, view, and modify file systems.

A file system consists of a set of disks that are used to store file metadata, data, and structures, including quota files and recovery logs. The file system achieves high-performance I/O in the following ways:

- Stripes data across multiple disks that are attached to multiple nodes:
 - All data in the file system is read and written in wide parallel stripes.
 - The data block size determines the maximum size of a read request or write request that a file system sends to the I/O device driver.
 - The block size, subblock size, and number of subblocks per block of a file system are set when the file system is created and cannot be changed later.
- Optimizes for small block write operations. A block is also subdivided into subblocks, so that multiple small block application writes can be aggregated and stored in a file system block, without wasting space in the block.
- Provides a high-performance metadata (inode) scan engine to scan the file system rapidly to enable fast identification of data that needs to be managed or migrated in the automated tiered storage environment.
- Supports a large block size that can be configured, when the file system is created, by the administrator to fit I/O requirements.
- Uses advanced algorithms that improve read-ahead and write-behind file functions for caching.
- Uses a sophisticated block-level locking based on a sophisticated token-management system that provides data consistency, while allowing multiple application nodes concurrent access to the files.

For more information, see *Block size in the IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Creating and deleting file system

Use the **Create File System** option available in the **Files > File Systems** page to create file systems on existing NSDs.

Deleting a file system removes all of the data on that file system. Use caution when performing this task. To delete a file system, select the file system to be deleted and then select **Delete** from the **Actions** menu.

File system monitoring options

The **File Systems** page provides an easy way to monitor the performance, health status, and configuration aspects of all the available file systems in the IBM Storage Scale cluster.

The following options are available to analyze the file system performance:

1. A quick view that gives the number of NSD servers and NSDs that are part of the available file systems that are mounted on the GUI server. It also provides overall capacity and total throughput details of these file systems. You can access this view by selecting the expand button that is placed next to the title of the page. You can close this view if not required.

The graphs that are displayed in the quick view are refreshed regularly. The refresh intervals depend on the displayed timeframe as shown:

- Every minute for a 5-minute interval
- Every 15 minutes for a 1-hour interval
- Every 6 hours for a 24-hour interval
- Every 2 days for a 7-day interval
- Every 7 days for a 30-day interval
- Every 4 months for a 365-day interval

2. A file system table that displays health status, performance details, and other important configuration aspects of file systems available in the system. The following important details are available in the file system table:

- File systems configured in the system
- Health status. The detailed information about the events reported against each file system are available in the **Events** tab of the file system detailed view.
- Capacity information
- Certain information of remote file systems that are mounted from a remote cluster.
- Mount status, mount configuration, and number of local and remote mounts.
- Number of pools that are part of the file system. A file system consists of one or more pools. Detailed information of pools of a file system is available in the **Pools** tab of the file system detailed view.
- Number of NSDs that are part of the file system. Detailed information of NSDs of a file system are available in the **NSDs** tab of the file system detailed view.
- Performance data. To find file systems with extreme values, you can sort the values that are displayed in the file systems table by different performance metrics. Click the performance metric in the table header to sort the data based on that metric. You can select the time range that determines the averaging of the values that are displayed in the table and the time range of the charts in the overview from the time range selector, which is placed in the upper right corner. The metrics in the table do not update automatically. The refresh button above the table allows to refresh the table with more recent data. The detailed performance details per node, pool, and NSDs are available in the detailed view of the file system.
- Protocols that are used to export or share the data that is stored in the file system.
- Number of nodes on which the file system is mounted. Details specific to each node on which the file system is mounted are available in the detailed view of the file system. You can also mount or unmount the file system from the detailed view.

3. A detailed view of the performance and health aspects of individual file systems. To see the detailed view, you can either double-click the file system for which you need to view the details or select the file system and click **View Details**.

The detailed performance view helps to drill down to various performance aspects. The following list provides the performance details that can be obtained from each tab of the performance view:

- **Overview:** Provides an overview of the file system performance.
- **Events:** System health events reported for the file system.

- **NSDs:** Details of the NSDs that are part of the file system.
- **Pools:** Details of the pools that are part of the file system.
- **Nodes:** Details of the nodes on which the file system is mounted.
- **Remote Nodes:** Details of the remote cluster nodes where the local file system is mounted.
- **File sets:** Details of the file sets that are part of the file system.
- **NFS:** Details of the NFS exports created in the file system.
- **SMB:** Details of the SMB shares created in the file system.
- **Object:** Details of the IBM Storage Scale object storage on the file system.
- **Properties:** Provides details of the file system attributes. You can also use the **Automatic mount** option to configure the automatic mount mode of the file system.

Managing access control

You can control the access to files and directories in a file system by defining access control lists (ACLs). ACLs can be inherited within a file system. The mount path of the file system does not inherit any ACL from a parent path. Therefore, you can set the ACL of the file system mount path using the **Edit Access Control** option.

When creating a file system, a default ACL is set. To modify the access controls defined for a file system, right-click the file system that is listed in the file system view and select **Edit Access Control**. The owner, owning group, and access control list cannot be modified if the directory is not empty. Users with the role *Dataaccess* are allowed to modify owner, group, and ACL even when the directory is not empty.

Mounting or unmounting a file system

You can use the IBM Storage Scale GUI to mount or unmount individual file systems or multiple file systems on the selected nodes. Use the **Files > File Systems**, **Files > File Systems > View Details > Nodes**, or **Nodes > View Details > File Systems** page in the GUI to mount or unmount a file system.

The GUI has the following options related to mounting the file system:

1. Mount local file systems on nodes of the local IBM Storage Scale cluster.
2. Mount remote file systems on local nodes.
3. Select individual nodes, protocol nodes, or nodes by node class while selecting nodes on which the file system needs to be mounted.
4. Prevent or allow file systems from mounting on individual nodes.

Do the following to prevent file systems from mounting on a node:

- a. Go to **Nodes**.
 - b. Select the node on which you need to prevent or allow file system mounts.
 - c. Select **Prevent Mounts** from the **Actions** menu.
 - d. Select the required option and click **Prevent Mount** or **Allow Mount** based on the selection.
5. Configure automatic mount option. The automatic configure option determines whether to automatically mount file system on nodes when GPFS daemon starts or when the file system is accessed for the first time. You can also specify whether to exclude individual nodes while enabling the automatic mount option. To enable automatic mount, do the following:
 - a. From the **Files > File Systems** page, select the file system for which you need to enable automatic mount.
 - b. Select **Configure Automatic Mount** option from the **Actions** menu.
 - c. Select the required option from the list of automatic mount modes.
 - d. Click **Configure**.

Note: You can configure automatic mount option for a file system only if the file system is unmounted from all nodes. That is, you need to stop I/O on this file system to configure this option. However, you can include or exclude the individual nodes for automatic mount without unmounting the file system from all nodes.

You can utilize the following options that are supported in the GUI to unmount file systems:

- Unmount local file system from local nodes and remote nodes.
- Unmount a remote file system from the local nodes. When a local file system is unmounted from the remote nodes, the remote nodes can no longer be seen in the GUI. The **Files > File Systems > View Details > Remote Nodes** page lists the remote nodes that currently mount the selected file system. The selected file system can be a local or a remote file system but the GUI permits to unmount only local file systems from the remote nodes.
- Select individual nodes, protocol nodes, or nodes by node class while selecting nodes from which the file system needs to be unmounted.
- Specify whether to force unmount. Selecting the **Force unmount option** while unmounting the file system unmounts the file system even if it is still busy in performing the I/O operations. Forcing the unmount operation affects the outstanding operations and causes data integrity issues. The IBM Storage Scale system relies on the native unmount command to carry out the unmount operation. The semantics of forced unmount are platform-specific. On certain platforms such as Linux, even when forced unmount is requested, file system cannot be unmounted if it is still referenced by system kernel. To unmount a file system in such cases, identify and stop the processes that are referencing the file system. You can use system utilities like *lsof* and *fuser* for this.

Some administrative actions like repairing file system structures by using the **mmfsck** command, require that the file system is unmounted on all nodes.

Policies

IBM Storage Scale provides a way to automate the management of files by using policies and rules. You can manage these policies and rules through the **Files > Information Lifecycle** page of the management GUI.

A policy rule is an SQL-like statement that tells the file system what to do with the data for a file in a specific pool if the file meets specific criteria. A rule can apply to any file being created or only to files being created within a specific fileset or group of filesets.

Managing file audit logging

File audit logging records file operations on a file system and logs them in a retention-enabled fileset. Each file operation is generated as a local event on the node that serves the file operation.

You can enable the file audit logging either while creating a file system by using the **Create File System** option or by using the **Enable File Auditing** option from the **Actions** menu for an already created file system.

While enabling the file audit logging, you can specify the following details:

- The file system in which the file audit log must be stored.
- Name of the fileset where the audit log must be stored.
- The period for which the log must be retained.

Note: The GUI offers to enable file audit logging for a file system if file audit logging is installed and configured. For more information on installing file audit logging and its components, see [Manually installing file audit logging](#).

To disable file audit logging, select the file system for which you need to disable the feature and select **Disable File Auditing** from the **Actions** menu.

To modify the list of file auditing events to be included in the log, select **Edit File Audit Logging** from the **Actions** menu.

For more information, see *File audit logging* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Chapter 27. File system format changes between versions of IBM Storage Scale

This topic describes features of IBM Storage Scale that operate only with file systems of a particular format level or higher.

Note: The features that are described in this topic are only a subset of the functional changes that are introduced with the different releases of IBM Storage Scale. Functional changes that do not require changing the file system format are not listed here. Such changes either are immediately available when the new version of IBM Storage Scale is installed, or you need to run the `mmchconfig release=LATEST` command after the installation.

Every IBM Storage Scale file system has a format level number that identifies the on-disk format of the file system and is an indicator of the supported file system functions. The following table summarizes which file system format level corresponds to each IBM Storage Scale 5.2.x.x or 5.1.x.x version.

Table 21. File system format changes between versions of IBM Storage Scale 5.2.x.x and 5.1.x.x	
IBM Storage Scale version	File system format level
5.2.1	35.00
5.2.0	34.00
5.1.9	33.00
5.1.8	31.00
5.1.7	
5.1.6	30.00
5.1.5	29.00
5.1.4	28.00
5.1.3	27.00
5.1.2	26.00
5.1.1	25.00
5.1.0	24.00

The file system format number is assigned when the file system is first created and can be updated to the latest supported level when the file system is migrated with the `mmchfs -V` command. The format number for a file system can be displayed with the `mmlsfs -V` command.

If a file system was created with an older IBM Storage Scale release, new functionality that requires different on-disk data structures is not enabled until you run the `mmchfs -V` command. Some new features might require you also to run the `mmigrate` command.

Note: The `-V` parameter cannot be used to create file systems that were created before GPFS 3.2.1.5 available to Windows nodes. Windows nodes can mount file systems only that were created with GPFS 3.2.1.5 or later.

The `mmchfs -V` parameter requires the specification of one of two values, `full` or `compat`:

- Specifying `mmchfs -V full` enables all of the new functionality that requires different on-disk data structures. After this command, nodes in remote clusters that are running an older GPFS version will no longer be able to mount the file system.

The **mmchfs -V full** command displays a warning as in the following example:

```
# mmchfs n03NsdoNfile36 -V full
You have requested that the file system be upgraded to
version 19.01 (5.0.1.0). This will enable new functionality but will
prevent you from using the file system with earlier releases of GPFS.
Do you want to continue?
```

- Specifying **mmchfs -V compat** enables only compatible with an earlier version format changes. Nodes in remote clusters that were able to mount the file system before the format changes can continue to do so afterward.

In IBM Storage Scale 5.2.1, new file systems are created at file system format level 35.00. To update a file system from an earlier format to format level 35.00, issue the following command:

```
mmchfs Device -V full
```

where *Device* is the device name of the file system. The following feature of IBM Storage Scale 5.2.1 requires a file system to be at format number 35.00 or later:

- A new operation can be logged as an event in the file audit logs. A CLOSEWRITE event indicates that a file was opened for writing and then closed. For more information, see the *File audit logging events and JSON attributes in file audit logging* in the *IBM Storage Scale: Problem Determination Guide*.

In IBM Storage Scale 5.2.0, new file systems are created at file system format level 34.00. To update a file system from an earlier format to format level 34.00, issue the following command:

```
mmchfs Device -V full
```

where *Device* is the device name of the file system. No feature of IBM Storage Scale 5.2.0 requires a file system to be at format number 34.00 or later.

In IBM Storage Scale 5.1.9, new file systems are created at file system format level 33.00. To update a file system from an earlier format to format level 33.00, issue the following command:

```
mmchfs Device -V full
```

where *Device* is the device name of the file system. No feature of IBM Storage Scale 5.1.9 requires a file system to be at format number 33.00 or later.

In IBM Storage Scale 5.1.8, new file systems are created at file system format level 31.00, which is the same that is used in IBM Storage Scale 5.1.7.

In IBM Storage Scale 5.1.7, new file systems are created at file system format level 31.00. To update a file system from an earlier format to format level 31.00, issue the following command:

```
mmchfs Device -V full
```

where *Device* is the device name of the file system. The following features of IBM Storage Scale 5.1.7 require a file system to be at format number 31.00 or later:

- A new option to disable AFM fileset without using the unlink method.
- A new option to enable **afmSyncReadMount** which creates a separate mount path at the gateway node through which synchronous read calls are sent to the home separately.

For more information, see the **mmchffilesset** and **mmcrffilesset** commands in *IBM Storage Scale: Command and Programming Reference Guide*.

In IBM Storage Scale 5.1.6, new file systems are created at file system format level 30.00. To update a file system from an earlier format to format level 30.00, issue the following command:

```
mmchfs Device -V full
```

where *Device* is the device name of the file system. The following feature of IBM Storage Scale 5.1.6 requires a file system to be at format number 30.00 or later:

- IBM Storage Scale supports FCM (FlashCore Module) 3.0 drives starting from IBM Storage Scale 5.1.6.

For more information, see *IBM Storage Scale with FCM (FlashCore Module) drives* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

In IBM Storage Scale 5.1.5, new file systems are created at file system format level 29.00. To update a file system from an earlier format to format level 29.00, issue the following command:

```
mmchfs Device -V full
```

where *Device* is the device name of the file system. The following features of IBM Storage Scale 5.1.5 require a file system to be at format number 29.00 or later:

- An option `--nfs4-owner-write-acl` is added to the **mmchfs**, **mmcdfs**, and **mmclsfs** commands to specify or display whether object owners are given implicit NFSv4 WRITE_ACL permission.

For more information, see the **mmchfs**, **mmcdfs**, and **mmclsfs** commands in *IBM Storage Scale: Command and Programming Reference Guide*.

- An option `--expiration-time` is added to the **mmcrsnapshot** command to specify the expiration time of a snapshot for which the retention period is defined.

For more information, see the **mmcrsnapshot** command in *IBM Storage Scale: Command and Programming Reference Guide*.

In IBM Storage Scale 5.1.4, new file systems are created at file system format level 28.00. To update a file system from an earlier format to format level 28.00, issue the following command:

```
mmchfs Device -V full
```

where *Device* is the device name of the file system. The following features of IBM Storage Scale 5.1.4 require a file system to be at format number 28.00 or later:

- Specifying the `--auto-inode-limit` option in the **mmcdfs** command automatically increases the maximum number of inodes per inode space in the file system.

For more information, see the **mmcdfs** command in *IBM Storage Scale: Command and Programming Reference Guide*.

- Allowing access to remote cluster nodes to only a subset of filesets instead of the entire file system.

For information on GPFS fileset access, see “[Fileset access control for remote clusters](#)” on page 515.

In IBM Storage Scale 5.1.3, new file systems are created at file system format level 27.00. To update a file system from an earlier format to format level 27.00, issue the following command:

```
mmchfs Device -V full
```

where *Device* is the device name of the file system. The following features of IBM Storage Scale 5.1.3 require a file system to be at format number 27.00 or later:

- Automatic flushing of disk buffers when closing files that were opened for write operations on the device.
- Extending quota management support to increase the maximum number of inodes in quota limits from 31 to 63 bits.

In IBM Storage Scale 5.1.2, new file systems are created at file system format level 26.00. To update a file system from an earlier format to format level 26.00, issue the following command:

```
mmchfs Device -V full
```

where *Device* is the device name of the file system. The following feature of IBM Storage Scale 5.1.2 requires a file system to be at format number 26.00 or later:

- Fine-Grained Directory Locking (FGDL) for link. FGDL is enabled to improve performance of link creation when concurrent hard links are created in a single directory from multiple nodes.

In IBM Storage Scale 5.1.1, new file systems are created at file system format level 25.00. To update a file system from an earlier format to format level 25.00, issue the following command:

```
mmchfs Device -V full
```

where *Device* is the device name of the file system. The following feature of IBM Storage Scale 5.1.1 requires a file system to be at format number 25.00 or later:

- The integrated archive management (IAM) mode support for the AFM-DR filesets.

On a primary site and the secondary site, the cluster version must be upgraded to the latest version by using the **mmchconfig release=LATEST** command.

In IBM Storage Scale 5.1.0, new file systems are created at file system format level 24.00. To update a file system from an earlier format to format level 24.00, issue the following command:

```
mmchfs Device -V full
```

where *Device* is the device name of the file system. The following feature of IBM Storage Scale 5.1.0 requires a file system to be at format number 24.00 or later:

- Creating AFM to cloud object storage filesets.

In IBM Storage Scale 5.0.5, new file systems are created at file system format level 23.00. To update a file system from an earlier format to format level 23.00, issue the following command:

```
mmchfs Device -V full
```

where *Device* is the device name of the file system. The following feature of IBM Storage Scale 5.0.5 requires a file system to be at format number 23.00 or later:

- Enabling FastCreate on AFM and AFM-DR filesets.

In IBM Storage Scale 5.0.4, new file systems are created at file system format level 22.00. To update a file system from an earlier format to format level 22.00, issue the following command:

```
mmchfs Device -V full
```

where *Device* is the device name of the file system. The following features of IBM Storage Scale 5.0.4 require a file system to be at format number 22.00 or later:

- Support for thin provisioned storage devices and NVMe SSDs.
- Support for linking GPFS dependent filesets inside AFM and AFM-DR filesets.

In IBM Storage Scale 5.0.3, new file systems are created at file system format level 21.00. To update a file system from an earlier format to format level 21.00, issue the following command:

```
mmchfs Device -V full
```

where *Device* is the device name of the file system. The following feature of IBM Storage Scale 5.0.3 requires a file system to be at format number 21.00 or later:

- Genomic compression

In IBM Storage Scale 5.0.2, new file systems are created at file system format number 20.01. To update a file system from an earlier format to format number 20.01, issue the following command:

```
mmchfs Device -V full
```

where *Device* is the device name of the file system. The following features of IBM Storage Scale 5.0.2 require a file system to be at format number 20.01 or later:

- The **afmGateway** attribute of the **mmchfileset** command specifies a user-defined gateway node for an AFM or AFM DR fileset that is given preference over the internal hashing algorithm.
- The **maxActiveIallocSegs** performance attribute of the **mmchconfig** command controls the number of active inode allocation segments that are maintained on a node. In IBM Storage Scale 5.0.2

and later the default number is 8 and the range is 1 - 64. In earlier versions the default value and also the maximum value is 1.

- The clustered watch folder feature provides you the ability to watch file operations across clusters. For more information, see the topic *Introduction to clustered watch folder* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

In IBM Storage Scale 5.0.1, new file systems are created at format number 19.01. To update the format of an earlier file system to format number 19.01, issue the following command:

```
mmchfs Device -V full
```

where *Device* is the device name of the earlier file system.

In IBM Storage Scale 5.0.0, new file systems are created at format number 18.00. To update the format of an earlier file system to format number 18.00, issue the following command:

```
mmchfs Device -V full
```

where *Device* is the device name of the earlier file system. The following features of IBM Storage Scale 5.0.0 require a file system to be at format number 18.00 or later:

- Smaller subblock sizes for file systems that have a large data block size.

Note: This feature is supported only for file systems that are created at file system format number 18.00 or later. It is not supported for file systems that are updated to format number 18.00 or later from an earlier format number. For more information, see the parameter **-B BlockSize** in the topic *mmcrfs* in the *IBM Storage Scale: Command and Programming Reference Guide*.

- File compression with the lz4 compression library
- File audit logging

In IBM Storage Scale 4.2.3.0, new file systems are created at format number 17.00. To update the format of an earlier file system to format number 17.00, issue the following command:

```
mmchfs Device -V full
```

where *Device* is the device name of the earlier file system. The following features of IBM Storage Scale 4.2.3.0 require a file system to be at format number 17.00 or later:

- Quality of Service for I/O (QoS)
- File compression with zlib compression library
- Information lifecycle management (ILM) for snapshots

If your current file system is at format number 14.20 (IBM Storage Scale 4.1.1), the set of enabled features depends on the value that is specified with the *mmchfs -V* option:

- After running *mmchfs -V full*, the file system can support the following:
 - Enabling and disabling of quota management without unmounting the file system.
 - The use of fileset-level integrated archive manager (IAM) modes.
- There are no new features that can be enabled with *mmchfs -V compat*.

If your current file system is at format number 14.04 (GPFS 4.1.0.0), the set of enabled features depends on the value specified with the *mmchfs -V* option:

- After running *mmchfs -V full*, the file system can support different block allocation map types on an individual storage-pool basis.
- There are no new features that can be enabled with *mmchfs -V compat*.

If your current file system is at format number 13.23 (GPFS 3.5.0.7), the set of enabled features depends on the value that is specified with the *mmchfs -V* option:

- After running *mmchfs -V full*, the file system can support the following:

- Directory block sizes can be up to 256 KB (previous maximum was 32 KB).
- Directories can reduce their size when files are removed.
- There are no new features that can be enabled with `mmchfs -V compat`.

If your current file system is at format number 13.01 (GPFS 3.5.0.1), the set of enabled features depends on the value specified with the `mmchfs -V` option:

- After running `mmchfs -V full`, the file system can support the following:
 - Extended storage pool properties
 - File Placement Optimizer (FPO)
 - Storing small directories in the inode
 - Storing the data for small files in the inode
- There are no new features that can be enabled with `mmchfs -V compat`.

If your current file system is at format number 12.03 (GPFS 3.4), the set of enabled features depends on the value specified with the `mmchfs -V` option:

- After running `mmchfs -V full`, the file system can support the following:
 - Independent filesets and snapshots of individual independent filesets
 - Active file management (AFM)
 - File clones (writable snapshots of a file)
 - Policy language support for new attributes, variable names, and functions: OPTS clause for the SET POOL and RESTORE rules, encoding of path names via an ESCAPE clause for the EXTERNAL LIST and EXTERNAL POOL rules, GetEnv(), GetMMconfig(), SetXattr(), REGEX().
- There are no new features that can be enabled with `mmchfs -V compat`.

If your current file system is at format number 11.03 (GPFS 3.3), the set of enabled features depends on the value specified with the `mmchfs -V` option:

- After running `mmchfs -V full`, the file system can support the following:
 - more than 2,147,483,648 files
 - fast extended attributes (which requires `mmigratefs` to be run also)
- There are no new features that can be enabled with `mmchfs -V compat`.

If your current file system is at format number 10.00 (GPFS 3.2.0.0) or 10.01 (GPFS 3.2.1.5), after running `mmchfs -V`, the file system can support all of the features included with earlier levels, plus the following:

- new maximum number of filesets in a file system (10000)
- new maximum for the number of hard links per object (2^{**32})
- improved quota performance for systems with large number of users
- policy language support for new attributes, variable names, and functions: MODE, INODE, NLINK, RDEVICE_ID, DEVICE_ID, BLOCKSIZE, GENERATION, XATTR(), ATTR_INTEGER(), and XATTR_FLOAT()

If your current file system is at format number 9.03 (GPFS 3.1), after running `mmchfs -V`, the file system can support all of the features included with earlier levels, plus:

- fine grain directory locking
- LIMIT clause on placement policies

If your current file system is at format number 8.00 (GPFS 2.3), after running `mmchfs -V`, the file system can support all of the features included with earlier levels, plus:

- Storage pools
- Filesets
- Fileset quotas

If your current file system is at format number 7.00 (GPFS 2.2), after running `mmchfs -V`, the file system can support all of the features that are included with earlier levels, plus:

- NFS V4 access control lists
- New format for the internal allocation summary files

If your current file system is at format number 6.00 (GPFS 2.1), after running `mmchfs -V`, the file system can support all of the features that are included with earlier levels and extended access control list entries (-rwxr access mode bits).

Chapter 28. Managing disks

Use the given information to manage disks in IBM Storage Scale.

Disk can have connectivity to each node in the cluster, be managed by network shared disk servers, or a combination of the two. For more information, see *mmlcrnsd command* in the *IBM Storage Scale: Command and Programming Reference Guide*. Also see, *Network Shared Disk (NSD) creation considerations* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Note: A LUN provided by a storage subsystem is a disk for the purposes of this documentation, even if the LUN is made up of multiple physical disks.

The disk related tasks performed on a GPFS file system include:

- [“Displaying disks in a GPFS cluster” on page 275](#)
- [“Adding disks to a file system” on page 276](#)
- [“Deleting disks from a file system” on page 276](#)
- [“Replacing disks in a GPFS file system” on page 278](#)
- [“Additional considerations for managing disks” on page 279](#)
- [“Displaying GPFS disk states” on page 280](#)
- [“Changing GPFS disk states and parameters” on page 281](#)
- [“Changing your NSD configuration” on page 283](#)
- [“Changing NSD server usage and failback” on page 284](#)
- [“Enabling and disabling Persistent Reserve” on page 284](#)

Displaying disks in a GPFS cluster

You can display the disks that belong to your GPFS cluster by issuing the **mmlsnsd** command.

The default is to display information for all disks defined to the cluster (-a). Otherwise, you may choose to display the information for a particular file system (-f) or for all disks which do not belong to any file system (-F).

To display the default information for all of the NSDs belonging to the cluster, enter:

```
mmlsnsd
```

The system displays information similar to:

File system	Disk name	NSD servers
fs2	hd3n97	c5n97g.ppd.pok.ibm.com,c5n98g.ppd.pok.ibm.com,c5n99g.ppd.pok.ibm.com
fs2	hd4n97	c5n97g.ppd.pok.ibm.com,c5n98g.ppd.pok.ibm.com,c5n99g.ppd.pok.ibm.com
fs2	hd5n98	c5n98g.ppd.pok.ibm.com,c5n97g.ppd.pok.ibm.com,c5n99g.ppd.pok.ibm.com
fs2	hd6n98	c5n98g.ppd.pok.ibm.com,c5n97g.ppd.pok.ibm.com,c5n99g.ppd.pok.ibm.com
fs2	sdbnsd	c5n94g.ppd.pok.ibm.com,c5n96g.ppd.pok.ibm.com
fs2	sdcnsd	c5n94g.ppd.pok.ibm.com,c5n96g.ppd.pok.ibm.com
fs2	sddnsd	c5n94g.ppd.pok.ibm.com,c5n96g.ppd.pok.ibm.com
fs2	sdensd	c5n94g.ppd.pok.ibm.com,c5n96g.ppd.pok.ibm.com
fs2	sdgnsd	c5n94g.ppd.pok.ibm.com,c5n96g.ppd.pok.ibm.com
fs2	sdfnsd	c5n94g.ppd.pok.ibm.com,c5n96g.ppd.pok.ibm.com
fs2	sdhnsd	c5n94g.ppd.pok.ibm.com,c5n96g.ppd.pok.ibm.com
(free disk)	hd2n97	c5n97g.ppd.pok.ibm.com,c5n98g.ppd.pok.ibm.com

To find out the local device names for the disks, use the **mmlsnsd** command with the -m option. For example, issuing **mmlsnsd -m** produces output similar to this:

Disk name	NSD volume ID	Device	Node name	Remarks
hd2n97	0972846145C8E924	/dev/hdisk2	c5n97g.ppd.pok.ibm.com	server node

hd2n97	0972846145C8E924	/dev/hdisk2	c5n98g.ppd.pok.ibm.com	server node
hd3n97	0972846145C8E927	/dev/hdisk3	c5n97g.ppd.pok.ibm.com	server node
hd3n97	0972846145C8E927	/dev/hdisk3	c5n98g.ppd.pok.ibm.com	server node
hd4n97	0972846145C8E92A	/dev/hdisk4	c5n97g.ppd.pok.ibm.com	server node
hd4n97	0972846145C8E92A	/dev/hdisk4	c5n98g.ppd.pok.ibm.com	server node
hd5n98	0972846245EB501C	/dev/hdisk5	c5n97g.ppd.pok.ibm.com	server node
hd5n98	0972846245EB501C	/dev/hdisk5	c5n98g.ppd.pok.ibm.com	server node
hd6n98	0972846245DB3AD8	/dev/hdisk6	c5n97g.ppd.pok.ibm.com	server node
hd6n98	0972846245DB3AD8	/dev/hdisk6	c5n98g.ppd.pok.ibm.com	server node

Adding disks to a file system

Many file systems grow rapidly, so after creating a file system you might decide that more disk space is required.

Storage in a file system is divided in storage pools. The maximum size of any one disk that can be added to an existing storage pool is set approximately to the sum of the disk sizes when the storage pool is created. The actual value is shown in the **mmdf** command output.

Once a storage pool is created, the maximum size *cannot* be altered. However, you can create a new pool with larger disks, and then move data from the old pool to the new one.

When establishing a storage pool and when adding disks later to an existing storage pool, you should try to keep the sizes of the disks fairly uniform. GPFS allocates blocks round robin, and as the utilization level rises on all disks. The small ones fill up first and all files created after that are spread across fewer disks, which reduces the amount of prefetch that can be done for those files.

To add disks to a GPFS file system, first decide if you will:

1. Create new disks using the **mmcrnsd** command.

In this case, you must also decide whether to create a new set of NSD and pools stanzas or use the rewritten NSD and pool stanzas that the **mmcrnsd** command produces. In a rewritten file, the disk usage, failure group, and storage pool values are the same as the values that are specified in the **mmcrnsd** command.

2. Select disks no longer in use in any file system. Issue the **mm1snsd -F** command to display the available disks.

The disk can then be added to the file system using the stanza file as input to the **mmadddisk** command.

Note: Rebalancing of files is an I/O intensive and time-consuming operation, and is important only for file systems with large files that are mostly invariant. In many cases, normal file update and creation will rebalance your file system over time, without the cost of the rebalancing.

For more information, see the *mmadddisk command* and the *mmcrnsd command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Deleting disks from a file system

Before you delete a disk from an IBM Storage Scale file system, use the **mmdf** command to determine whether the file system has sufficient free space to accommodate all data or metadata in the storage pool where the disk is located.

- See “[Querying file system space](#)” on page 241 for more information about diagnosing space problems.
- For file systems where the majority of files are smaller than the file system block size, resulting in fragmentation of storage space, it is recommended that the free space in the file system be at least 150 percent of the space consumed by the disks that will be deleted. For example, to delete a 400 GB disk from a file system, you must first confirm that the file system contains minimum 600 GB of free space in the storage pool from where the disk is being removed.

Note: Rebalancing of files is an I/O intensive and time consuming operation, and is important only for file systems with large files that are mostly invariant. In many cases, normal file update and creation rebalance your file system over time, without the cost of the rebalancing.

- A disk that is being deleted must be in an accessible state, that is started or suspended, to allow for data to be moved off of the disk to another disk in the storage pool. If a disk is being removed because it is damaged or is no longer working, then additional precautions are necessary before the disk is deleted from the file system. Deleting a disk that is no longer accessible can cause a loss of data, and if the disk contains metadata, you may need to run the **mmfsck** command. For further information about removing damaged or inaccessible disk see *NSD and underlying disk subsystem failures* and *Disk media failure* in the *IBM Storage Scale: Problem Determination Guide*.

When you need to remove or replace one or more disks in a file system, you can use the **mmrestripefs** command. Because this command is tolerant to stop and restart, you can control the process of moving data off of one or more disks that are being removed or replaced. You can also use the quality of service (QoS) feature (the **mmqos** command) to control the impact of the I/Os necessary to move data from one or more disks that are being removed or replaced to other disks.

To use the **mmrestripefs** command before the removal or replacement of a disk complete the following steps:

1. Confirm that there is sufficient free space in the storage pool where the disk is located.
2. Suspend the disks that need to be removed or replaced by using the **mmchdisk** command.
3. Use the **mmrestripefs** command to move all data from the suspended disks.
4. Use the **mmdeldisk** command to remove one or more disks, or use the **mmrpldisk** command to replace one or more disks.

For syntax and usage information, refer to *mmdeldisk command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Specify the file system and the names of one or more disks to delete with the **mmdeldisk** command. For example, to delete the disk **hd2n97** from the file system **fs2** enter:

```
mmdeldisk fs2 hd2n97
```

The system displays information similar to:

```
Deleting disks ...
Scanning system storage pool
Scanning file system metadata, phase 1 ...
19 % complete on Fri Mar 16 23:23:50 2012
100 % complete on Fri Mar 16 23:23:51 2012
Scan completed successfully.
Scanning file system metadata, phase 2 ...
46 % complete on Fri Mar 16 23:23:55 2012
93 % complete on Fri Mar 16 23:23:58 2012
100 % complete on Fri Mar 16 23:23:58 2012
Scan completed successfully.
Scanning file system metadata, phase 3 ...
Scan completed successfully.
Scanning file system metadata, phase 4 ...
Scan completed successfully.
Scanning user file metadata ...
19.50 % complete on Fri Mar 16 23:24:25 2012 ( 35777 inodes 1207 MB)
47.92 % complete on Fri Mar 16 23:24:49 2012 ( 199955 inodes 2966 MB)
50.05 % complete on Fri Mar 16 23:25:09 2012 ( 235356 inodes 3098 MB)
53.09 % complete on Fri Mar 16 23:25:31 2012 ( 261831 inodes 3286 MB)
55.12 % complete on Fri Mar 16 23:25:51 2012 ( 283815 inodes 3412 MB)
63.25 % complete on Fri Mar 16 23:26:12 2012 ( 319236 inodes 3915 MB)
63.27 % complete on Fri Mar 16 23:26:33 2012 ( 382031 inodes 6223 MB)
63.29 % complete on Fri Mar 16 23:27:03 2012 ( 699858 inodes 9739 MB)
100.00 % complete on Fri Mar 16 23:27:35 2012
Scan completed successfully.
Checking Allocation Map for storage pool 'system'
17 % complete on Fri Mar 16 23:27:42 2012
31 % complete on Fri Mar 16 23:27:47 2012
48 % complete on Fri Mar 16 23:27:52 2012
62 % complete on Fri Mar 16 23:27:57 2012
76 % complete on Fri Mar 16 23:28:02 2012
90 % complete on Fri Mar 16 23:28:07 2012
100 % complete on Fri Mar 16 23:28:08 2012
tsdeldisk completed.
mmdeldisk: Propagating the cluster configuration data to all
          affected nodes. This is an asynchronous process.
```

Replacing disks in a GPFS file system

Replacing an existing disk in a GPFS file system with a new one is the same as performing a delete disk operation followed by an add disk. However, this operation eliminates the need to restripe the file system following the separate delete disk and add disk operations as data is automatically moved to the new disk.

When replacing disks in a GPFS file system, first decide if you will:

1. Create new disks using the **mmcrnsd** command.

In this case, you must also decide whether to create a new set of NSD and pools stanzas or use the rewritten NSD and pool stanzas that the **mmcrnsd** command produces. In a rewritten file, the disk usage, failure group, and storage pool values are the same as the values that are specified in the **mmcrnsd** command.

2. Select NSDs no longer in use by another GPFS file system. Issue the **mm1snsd -F** command to display the available disks.

To replace a disk in the file system, use the **mmrpldisk** command. For example, to replace the NSD **hd3n97** in file system **fs2** with the existing NSD **hd2n97**, which is no longer in use by another file system, enter:

```
mmrpldisk fs2 hd3n97 hd2n97
```

The system displays information similar to:

```
Replacing hd3n97 ...

The following disks of fs2 will be formatted on node c33f2in01:
hd2n97: size 571398144 KB
Extending Allocation Map
Checking Allocation Map for storage pool 'system'
9 % complete on Fri Mar 16 23:33:29 2012
23 % complete on Fri Mar 16 23:33:34 2012
37 % complete on Fri Mar 16 23:33:40 2012
52 % complete on Fri Mar 16 23:33:45 2012
66 % complete on Fri Mar 16 23:33:50 2012
83 % complete on Fri Mar 16 23:33:55 2012
98 % complete on Fri Mar 16 23:34:00 2012
100 % complete on Fri Mar 16 23:34:00 2012
Completed adding disks to file system fs2.
Scanning system storage pool
Scanning file system metadata, phase 1 ...
13 % complete on Fri Mar 16 23:34:19 2012
100 % complete on Fri Mar 16 23:34:22 2012
Scan completed successfully.
Scanning file system metadata, phase 2 ...
29 % complete on Fri Mar 16 23:34:26 2012
67 % complete on Fri Mar 16 23:34:29 2012
100 % complete on Fri Mar 16 23:34:32 2012
Scan completed successfully.
Scanning file system metadata, phase 3 ...
Scan completed successfully.
Scanning file system metadata, phase 4 ...
Scan completed successfully.
Scanning user file metadata ...
8.21 % complete on Fri Mar 16 23:34:54 2012 ( 37741 inodes 770 MB)
22.65 % complete on Fri Mar 16 23:35:14 2012 ( 40182 inodes 2124 MB)
32.95 % complete on Fri Mar 16 23:35:34 2012 ( 160837 inodes 3090 MB)
35.15 % complete on Fri Mar 16 23:35:57 2012 ( 227991 inodes 3296 MB)
36.34 % complete on Fri Mar 16 23:36:17 2012 ( 265748 inodes 3408 MB)
37.34 % complete on Fri Mar 16 23:36:38 2012 ( 284398 inodes 3502 MB)
46.07 % complete on Fri Mar 16 23:37:04 2012 ( 310636 inodes 4320 MB)
61.41 % complete on Fri Mar 16 23:37:25 2012 ( 315141 inodes 5759 MB)
87.04 % complete on Fri Mar 16 23:37:50 2012 ( 350241 inodes 8163 MB)
87.06 % complete on Fri Mar 16 23:38:11 2012 ( 370562 inodes 10136 MB)
87.08 % complete on Fri Mar 16 23:38:42 2012 ( 392561 inodes 11982 MB)
87.10 % complete on Fri Mar 16 23:39:22 2012 ( 401049 inodes 13195 MB)
87.12 % complete on Fri Mar 16 23:40:14 2012 ( 1100590 inodes 15685 MB)
100.00 % complete on Fri Mar 16 23:40:57 2012
Scan completed successfully.
Checking Allocation Map for storage pool 'system'
10 % complete on Fri Mar 16 23:41:02 2012
26 % complete on Fri Mar 16 23:41:07 2012
```

```
33 % complete on Fri Mar 16 23:41:12 2012
44 % complete on Fri Mar 16 23:41:17 2012
68 % complete on Fri Mar 16 23:41:22 2012
100 % complete on Fri Mar 16 23:41:25 2012
Done
mmrpldisk: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

Note: If you attempt to replace a stopped disk and the file system is not replicated, the attempt will fail.

However, you can replace a stopped disk if the file system is replicated. You can do so in one of the following ways:

- Deletion, addition, and rebalancing method:

1. Use the **mmdealdisk** command to delete the stopped disk from the file system.
2. Use the **mmadddisk** command to add a replacement disk.
3. Use the **mmrestripefs -b** command to rebalance the file system.

While this method requires rebalancing, it returns the system to a protected state faster (because it can use all of the remaining disks to create new replicas), thereby reducing the possibility of losing data.

—Or—

- Direct replacement method:

Use the **mmrpldisk** command to directly replace the stopped disk.

The **mmrpldisk** command only runs at single disk speed because all data being moved must be written to the replacement disk. The data is vulnerable while the command is running, and should a second failure occur before the command completes, it is likely that some data will be lost.

For more information on handling this situation, see *Disk media failure* in the *IBM Storage Scale: Problem Determination Guide*.

Additional considerations for managing disks

Read additional considerations for managing disks.

Writing new data with strict replication when a disk is offline

In this scenario, strict replication is enforced for a file system and then a disk that is used by the file system is deleted, replaced, or suspended. If IBM Storage Scale then tries to create or append data to an existing file in the file system, the operation can fail with an ENOSPC error.

Note: To determine whether a file system has strict replication enforced, issue the following command:

```
mmlsfs fs1 -K
```

To write or append the data, follow these steps:

1. Disable strict replication.
2. Write the data.
3. Re-enable strict replication.
4. Issue the following command to migrate data off the suspended disk:

```
mmrestripefs fs1 -r
```

Displaying GPFS disk states

You can display the current state of one or more disks in your file system by issuing the `mmlsdisk` command.

The information includes parameters that were specified on the `mmcrfs` command, and the current availability and status of the disks. For example, to display the status of the disk `hd8vsdn100` in the file system `fs1`, enter:

```
mmlsdisk fs1 -d hd8vsdn100
```

Status is displayed in a format similar to:

disk name	driver type	sector size	failure group	holds metadata	holds data	status	availability	storage pool
hd8vsdn100	nsd	512	1 no	yes	ready	up		sp1

For syntax and usage information, see `mmlsdisk` command section in the *IBM Storage Scale: Command and Programming Reference Guide*.

Disk availability

The following information lists the possible values of disk availability, and what they mean.

A disk's availability determines whether GPFS is able to read and write to the disk. There are four possible values for availability:

up

The disk is available to GPFS for normal **read** and **write** operations.

down

No **read** and **write** operations can be performed on the disk.

recovering

An intermediate state for disks coming up, during which GPFS verifies and corrects data. A **write** operation can be performed while a disk is in this state, but a **read** operation cannot, because data on the disk being recovered might be stale until the `mmchdisk start` command completes.

unrecovered

The disk was not successfully brought up.

Disk availability is automatically changed from **up** to **down** when GPFS detects repeated I/O errors. You can also change the availability of a disk by issuing the `mmchdisk` command.

Related concepts

Disk status

The following information lists the possible values for disk status, and what they mean.

Disk status

The following information lists the possible values for disk status, and what they mean.

Disk status controls data placement and migration. Status changes as a result of a pending delete operation, or when the `mmchdisk` command is issued to allow file rebalancing or re-replicating prior to disk replacement or deletion.

Disk status has seven possible values, but four are transitional:

ready

Normal status.

suspended

or

to be emptied

Indicates that data is to be migrated off this disk.

being emptied

Transitional status in effect while a disk deletion is pending.

emptied

Indicates that data is already migrated off this disk.

replacing

Transitional status in effect for old disk while replacement is pending.

replacement

Transitional status in effect for new disk while replacement is pending.

GPFS allocates space only on disks with a status of **ready** or **replacement**.

GPFS migrates data off disks with a status of **being emptied**, **replacing**, **to be emptied**, or **suspended** onto disks with a status of **ready** or **replacement**. During disk deletion or replacement, data is automatically migrated as part of the operation. Issue the **mmrestripefs** command to initiate data migration from a suspended disk.

See “[Deleting disks from a file system](#)” on page 276, “[Replacing disks in a GPFS file system](#)” on page 278, and “[Restriping a GPFS file system](#)” on page 240.

Related conceptsDisk availability

The following information lists the possible values of disk availability, and what they mean.

Changing GPFS disk states and parameters

You might find it necessary to change a disk's state if there is some indication of disk failure or if you need to restripe the file system.

Refer to “[Displaying GPFS disk states](#)” on page 280 for a detailed description of disk states. You can change both the availability and status of a disk by using the **mmchdisk** command:

- Change disk availability by using the **mmchdisk** command and the **stop** and **start** options.
- Change disk status by using the **mmchdisk** command and the **suspend** and **resume** options.

Issue the **mmchdisk** command with one of the following four options to change disk state:

resume

Informs GPFS that a disk previously suspended is now available for allocating new space. Resume a disk only when you suspended it and decided not to delete or replace it. If the disk is in a stopped state, it remains stopped until you specify the **start** option. Otherwise, normal read and write access to the disk resumes.

start

Informs GPFS that a disk previously stopped is now accessible. GPFS does this by first changing the disk availability from down to recovering. The file system metadata is then scanned and any missing updates (replicated data that was changed while the disk was down) are repaired. If this operation is successful, the availability is then changed to up.

If the metadata scan fails, availability is set to unrecovered. This can occur when other disks remain in recovering or an I/O error occurred. Repair all disks and paths to disks. It is recommended to run **mmfsck** command. For more information, see **mmfsck** command in the *IBM Storage Scale: Command and Programming Reference Guide*. The metadata scan can be reinitiated later by issuing the **mmchdisk start** command again.

If more than one disk in the file system is down, they should all be started at the same time by using the **-a** option. If you start them separately and metadata is stored on any disk that remains down, the **mmchdisk start** command fails.

stop

Instructs GPFS to stop any attempts to access the specified disk. Use this option to inform GPFS that a disk failed or is inaccessible because of maintenance. A disk's availability remains down until it is explicitly started with the **start** option.

suspend
or
empty

Instructs GPFS to stop allocating space on the specified disk. Place a disk in this state before disk deletion or replacement. This is a user-initiated state that GPFS is never used without an explicit command to change disk state.

Note: A disk remains suspended until it is explicitly resumed. Restarting GPFS or rebooting nodes does not restore normal access to a suspended disk.

The empty option is similar to the suspend option. In GPFS 4.1.1 and earlier, the output of the **mmlsdisk** command displays the status as suspended, as shown in the following example.

For example, to suspend the *hd8vsdn100* disk in the file system *fs1*, enter:

```
mmchdisk fs1 suspend -d hd8vsdn100
```

To confirm the change, enter:

```
mmlsdisk fs1 -d hd8vsdn100
```

The system displays information similar to:

disk name	driver type	sector size	failure group	holds metadata	holds data	status	availability	storage pool
hd8vsdn100	nsd	512	7	yes	yes	suspended	up	system

For GPFS 4.1.1 and later, the status in the **mmlsdisk** command is displayed as to be emptied, as shown in the following example:

For example, to set to be emptied state for *gpfs1nsd* disk of the file system *fs1*, enter:

```
mmchdisk fs1 empty -d gpfs1nsd
```

To confirm the change, enter:

```
mmlsdisk fs1 -d gpfs1nsd
```

The system displays information similar to:

disk name	driver type	sector size	failure group	holds metadata	holds data	status	availability	disk id	storage pool
gpfs1nsd	nsd	512	-1	Yes	Yes	to be emptied	up	1	system
gpfs2nsd	nsd	512	-1	Yes	Yes	to be emptied	up	2	system
desc									

You can also use the **mmchdisk** command with the change option to change the *Disk Usage* and *Failure Group* parameters for one or more disks in a GPFS file system. This can be useful in situations where, for example, a file system that contains only RAID disks is being upgraded to add conventional disks that are better suited to storing metadata. After adding the disks by using the **mmadddisk** command, the metadata that is stored on the RAID disks must be moved to the new disks to achieve the desired performance improvement. To accomplish this, first the **mmchdisk change** command would be issued to change the *Disk Usage* parameter for the RAID disks to *dataOnly*. Then, the **mmrestripefs** command would be used to restripe the metadata off the RAID device and onto the conventional disks.

For example, to specify that metadata should no longer be stored on disk *hd8vsdn100*, enter:

```
mmchdisk fs1 change -d "hd8vsdn100:::dataOnly"
```

To confirm the change, enter:

```
mmlsdisk fs1 -d hd8vsdn100
```

The system displays information similar to:

disk name	driver type	sector size	failure group	holds metadata	holds data	status	availability	storage pool
hd8vsdn100	nsd	512	1 no	yes	ready	up	up	sp1

For more information, see the **mmchdisk** command and the **mmlsdisk** command in the *IBM Storage Scale: Command and Programming Reference Guide*.

Changing your NSD configuration

Use the following steps to change the NSD configuration.

Once your NSDs have been created, you may change the configuration attributes as your system requirements change. For more information about creating NSDs, see the *IBM Storage Scale: Concepts, Planning, and Installation Guide* and the **mmcrnsd** command in the *IBM Storage Scale: Command and Programming Reference Guide*.

By issuing the **mmchnsd** command you can:

- Specify up to eight servers for an NSD that does not have one.
- Change the NSD server nodes specified in the server list.
- Delete the server list. The disk must now be SAN-attached to all nodes in the cluster on which the file system will be mounted.

You must follow these rules when changing NSDs:

- Identify the disks by the NSD names that were given to them by the **mmcrnsd** command.
- Explicitly specify values for all NSD servers in the list, even if you are only changing one of the values.
- Connect the NSD to the new nodes prior to issuing the **mmchnsd** command.
- The **mmchnsd** command cannot change the *DiskUsage* or *FailureGroup* for an NSD. Use the **mmchdisk** command to change these attributes.
- To move a disk from one storage pool to another, use the **mmdealdisk** and **mmadddisk** commands.
- You cannot change the name of the NSD.

For example, to assign node k145n07 as an NSD server for disk gpfs47nsd:

- Make sure that k145n07 is not already assigned to the server list by issuing the **mmlsnsd** command.

```
mmlsnsd -d "gpfs47nsd"
```

The system displays information similar to:

File system	Disk name	NSD server nodes
fs1	gpfs47nsd	k145n09

- Ensure that the disk is connected to the new node k145n07.

- Issue the **mmchnsd** command:

```
mmchnsd "gpfs47nsd:k145n09,k145n07"
```

- Verify the changes by issuing the **mmlsnsd** command:

```
mmlsnsd -d gpfs47nsd
```

The system displays information similar to:

File system	Disk name	NSD servers
fs2	gpfs47nsd	k145n09.ppd.pok.ibm.com,k145n07.ppd.pok.ibm.com

Changing NSD server usage and fallback

GPFS determines if a node has physical or virtual connectivity to an underlying NSD disk through a sequence of commands invoked from the GPFS daemon. This determination is called disk discovery and occurs at both initial GPFS startup as well as whenever a file system is mounted.

The default order of access used in disk discovery:

1. Local block device interfaces for SAN, SCSI or IDE disks
2. NSD servers

The `useNSDserver` file system mount option can be used to set the order of access used in disk discovery, and limit or eliminate switching from local access to NSD server access, or the other way around. This option is specified using the `-o` flag of the `mmount`, `mount`, `mmchfs`, and `mmremotefs` commands, and has one of these values:

always

Always access the disk using the NSD server.

asfound

Access the disk as found (the first time the disk was accessed). No change of disk access from local to NSD server, or the other way around, is performed by GPFS.

asneeded

Access the disk any way possible. This is the default.

never

Always use local disk access.

For example, to always use the NSD server when mounting file system `fs1`, issue this command:

```
mmount fs1 -o useNSDserver=always
```

To change the disk discovery of a file system that is already mounted: cleanly unmount it, wait for the unmount to complete, and then mount the file system using the desired `-o useNSDserver` option.

NSD servers: Periodic checks for I/O problems

NSD servers periodically check disk reads on locally attached disks to detect disk problems early rather than wait until an I/O error occurs.

The periodic check occurs every five minutes by default. The periodic checks begin when the NSD server is created and continue until the NSD server is deleted.

If a problem is found, the `diskIOErr` event is triggered. For more information, see `mmaddcallback` command in the *IBM Storage Scale: Command and Programming Reference Guide*.

Enabling and disabling Persistent Reserve

GPFS can use Persistent Reserve (PR) functionality to improve failover times (with some restrictions).

The following restrictions apply to the use of PR:

- PR is supported on both AIX and Linux nodes. However, note the following:
 - If the disks have defined NSD servers, then the NSD server nodes must all be running AIX, or they must all be running Linux.
 - If the disks are SAN-attached to all nodes, then the SAN-attached nodes in the cluster must all be running AIX, or they must all be running Linux.
- The disk subsystems must support PR
- GPFS supports a mix of PR disks and other disks. However, you will only realize improved failover times if **all** disks in the cluster support PR.

- GPFS only supports PR in the local cluster. Remote mounts must access the disks through an NSD server.
- When you enable or disable PR, you must stop GPFS on all nodes.
- Before enabling PR, make sure all disks are in the same initial state.
- Before enabling PR, disks must be removed from any CCR tiebreaker disk configuration. After the enablement is finished, disks can be added back to the CCR tiebreaker disk configuration.

To enable Persistent Reserve, enter the following command:

```
mmchconfig usePersistentReserve=yes
```

To disable Persistent Reserve, enter the following command:

```
mmchconfig usePersistentReserve=no
```

For fast recovery times with Persistent Reserve, you should also set the *failureDetectionTime* configuration parameter. For fast recovery, a recommended value would be 10. You can set this by issuing the command:

```
mmchconfig failureDetectionTime=10
```

To determine if the disks on the servers and the disks of a specific node have PR enabled, issue the following command from the node:

```
mmlsnsd -X
```

The system responds with something similar to:

Disk name	NSD volume ID	Device	Devtype	Node name	Remarks
gpfs10nsd	09725E5E43035A99	/dev/hdisk6	hdisk	k155n14.kgn.ibm.com	server node,pr=yes
gpfs10nsd	09725E5E43035A99	/dev/hdisk8	hdisk	k155n16.kgn.ibm.com	server node,pr=yes
gpfs10nsd	09725E5E43035A99	/dev/hdisk6	hdisk	k155n17.kgn.ibm.com	directly attached pr=yes

If the GPFS daemon has been started on all the nodes in the cluster and the file system has been mounted on all nodes that have direct access to the disks, then pr=yes should be on all hdisks. If you do not see this, there is a problem. Refer to the *IBM Storage Scale: Problem Determination Guide* for additional information on Persistent Reserve errors.

Chapter 29. Managing protocol services

GPFS provides system administrators with the ability to manage the protocol services such as NFS, SMB, and object services.

For information on the CES HDFS protocol, see [CES HDFS](#).

Configuring and enabling SMB, S3, and NFS protocol services

If you did not previously enable and start the Cluster Export Services (CES) protocol services, enable and start them now.

Prerequisites

When you enable SMB protocol services, the following prerequisites must be met:

- The number of CES nodes must be 16 or lower.
- All CES nodes must be running the same system architecture. For example, mixing nodes based on Intel and Power is not supported.
- A valid **mmuserauth config** command.

When you add new CES nodes to a running system where the SMB protocol is enabled, the following prerequisite must be met:

- All SMB packages (gpfs.smb) must have the same version.
- All CES nodes must be in SMB **HEALTHY** state. You can verify the health status of the SMB service by using the **mmces state show smb** command.

When you remove a CES node from a running system where the SMB protocol is enabled, the following prerequisite must be met:

- All CES nodes (except for the node that is being removed) must be in SMB **HEALTHY** state.

For more information about the SMB states, see **mmces command** in *IBM Storage Scale: Command and Programming Reference Guide*.

Before enabling S3 services, see the *Planning for S3* section in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Enabling protocol services

Issue the following commands to enable SMB, S3, and NFS services on all CES nodes:

- **mmces service enable SMB**
- **mmces service enable NFS**
- **mmces service enable S3**

GUI navigation

- To enable SMB services in the GUI, log on to the IBM Storage Scale GUI and select **Services > SMB**.
- To enable NFS services in the GUI, log on to the IBM Storage Scale GUI and select **Services > NFS**.

The protocol services that are used need to be started on all CES nodes:

```
mmces service start SMB -a  
mmces service start NFS -a
```

```
mmces service start S3 -a
```

After you start the protocol services, verify that they are running by issuing the **mmces state show** command.

Note: The start and stop are maintenance commands. Stopping a service on a particular protocol node without first suspending the node ensures that the public IP addresses on that node stay with that node. In this case, protocol clients that try to connect to the service with these IP addresses fail. The NFS service might restart automatically after downtime if the process had shutdown unexpectedly.

The following example demonstrates how to manage NFS service, exports, and authentication options:

1. Issue the following command to enable the service:

```
mmces service enable NFS
```

Note: This command also starts NFS on all CES nodes.

2. Set up the authentication method. The following command specifies the **file** data access method and the **userdefined** authentication type:

```
mmuserauth service create --data-access-method file --type userdefined
```

3. Issue the following command to add an export:

```
mmnfs export add /gpfs/fs0/fset0
```

where **fs0** is a GPFS file system and **fset0** is an independent fileset.

4. Issue the following commands to verify that the service is configured and running:

```
mmces service list -a  
mmuserauth service list  
mmnfs export list
```

5. Issue the following commands to stop NFS and disable the NFS protocol on the CES nodes:

```
mmces service stop nfs -a  
mmuserauth service remove --data-access-method file
```

Note: The sequence for removing the **file** data access method is different for NFS and SMB. For NFS, you must remove the **file** data access method before you disable NFS.

6. Issue the following command to disable the NFS service on the CES nodes:

```
mmces service disable NFS
```

Important: When you disable NFS, the NFS configuration is lost. To save the NFS configuration, back up the contents of the `/var/mmfs/ces/nfs-config/` directory on any protocol node.

Support of `vfs_fruit` for the SMB protocol

The `vfs_fruit` module of the IBM Storage Scale SMB server provides enhanced compatibility for Apple SMB2 clients by implementing a set of SMB2 protocol extensions that are added by Apple to their SMB client and server.

These extensions help to increase the browsing speed in the Apple Finder application on network shares by enhancing directory listings and Apple metadata handling. This metadata is Apple-specific information on files like coloring for example (in contrast to file system metadata like i-nodes, timestamps that are not affected). Without these extensions Mac SMB2 clients store their Apple file metadata in accompanying files for each file (resource fork, file names that are starting with “`._`”).

The `vfs_fruit` module is disabled by default. The `vfs_fruit` module must be enabled to improve the performance of Apple file system. You must take precise decision to enable the `vfs_fruit` module

because disabling the module again requires an extended downtime to find all the files with fruit xattrs and converting the files into apple-double files.

The `vfs_fruit` module relies on Alternate Data Streams (ADS) support of a share. With IBM Storage Scale, the alternate data streams are stored in the extended attributes in the file system. As Mac clients need to see a consistent file server that either supports these extensions or not, `vfs_fruit` (and consequently stream) support is enabled globally and it cannot be configured on the share level. This means that Windows clients also notice that the IBM Storage Scale SMB2 server offers support of streams. As streams are stored in extended attributes certain size limitations apply, a single stream can be at most 16 KB and all streams in total can be 50 KB. Applications trying to write beyond those limits receive the error `NTSTATUS_DISK_FULL`.

The module intercepts the OS X special streams "AFP_AfpInfo" and "AFP_Resource" and handles them in a special way. Moreover, it enhances directory listings with Apple-specific metadata to reduce the number of network round trips on SMB2 find requests.

If files on a file server were accessed previously by Mac clients over SMB2, the Apple file metadata is already written to the accompanying files. Enabling fruit finds and reads this metadata, moves it into streams (extended attributes) and removes the accompanying files on file access. Thus, the metadata that is written earlier is not lost but it can take some time until all metadata is converted. Creating new files use streams (extended attributes) directly.

Disabling `vfs_fruit` support requires a conversion of the Apple-specific file metadata back from extended attributes to resource fork files to avoid losing metadata. For more information, see the *Disabling vfs_fruit support topic*, which is provided later in this section.

Enabling `vfs_fruit` support

The module can be activated globally by using the `mmsmb` command. To enable it consistently for all clients, the SMB service must be stopped on all CES nodes. Trying to enable it while SMB is running leads to the following error message:

```
[root@gpfs-11 rhel7]# mmsmb config change --vfs-fruit-enable
SMB Environment check failed.
Required: SMB service is not running on any CES node
Detected: SMB service is running on at least one CES node
```

Here is how to stop SMB on all CES nodes. If NFS is also enabled and has an AD dependency, then NFS needs to be stopped first.

```
[root@gpfs-11 rhel7]# mmces service stop smb -N cesnodes
gpfs-11.novalocal: SMB: service already stopped.
gpfs-12.novalocal: Redirecting to /bin/systemctl stop gpfs-smb.service
gpfs-12.novalocal: SMB: service successfully stopped.
gpfs-11.novalocal: CTDB: service already stopped.
gpfs-12.novalocal: Redirecting to /bin/systemctl stop gpfs-ctdb.service
gpfs-12.novalocal: CTDB: service successfully stopped.
```

To check whether SMB is running, issue the following command:

```
[root@gpfs-11 rhel7]# mmces service list -N cesnodes
Enabled services: SMB
gpfs-11.novalocal: SMB is not running
gpfs-12.novalocal: SMB is not running
```

Now that SMB is stopped, you can enable `vfs_fruit`:

```
[root@gpfs-11 rhel7]# mmsmb config change --vfs-fruit-enable
WARNING: You are about to enable the vfs_fruit module. It is not possible
to disable this module again without contacting IBM support first.

Are you sure you want to continue?
Enter "yes" or "no": yes
vfs_fruit enabled.
```

Now, the SMB service can be restarted:

```
[root@gpfs-11 rhel7]# mmces service start SMB -N cesnodes
gpfs-12.novalocal: Redirecting to /bin/systemctl start gpfs-ctdb.service
gpfs-11.novalocal: Redirecting to /bin/systemctl start gpfs-ctdb.service
gpfs-12.novalocal: CTDB: service successfully started.
gpfs-11.novalocal: CTDB: service successfully started.
gpfs-12.novalocal: Wait for ctdb to become ready. State=STARTUP
gpfs-11.novalocal: Wait for ctdb to become ready. State=STARTUP
gpfs-12.novalocal: Redirecting to /bin/systemctl start gpfs-smb.service
gpfs-11.novalocal: Redirecting to /bin/systemctl start gpfs-smb.service
gpfs-12.novalocal: SMB: service successfully started.
gpfs-11.novalocal: SMB: service successfully started.
```

Disabling vfs_fruit support

Enabling the `vfs_fruit` module changes how the Apple file metadata is stored, namely in extended attributes rather than in accompanying files. Thus, disabling `vfs_fruit` causes the Apple file metadata not to be used or found anymore. If `vfs_fruit` really needs to be disabled on a cluster after Apple file metadata, which should not be lost, is written, then contact IBM support for ways how to move Apple file metadata back out of the extended attributes into resource fork files. This procedure involves running a GPFS policy and a tool to move Apple-specific metadata that is changed by Mac OS clients from extended attributes back to resource fork files. This requires an SMB downtime.

Verifying vfs_fruit running status

To verify if the `vfs_fruit` module is enabled, issue the following command:

```
mmsmb config change --vfs-fruit-enable
```

After you issue the command, if the `vfs_fruit` module is enabled, you get a message as shown in the following sample output:

SMB Module `vfs_fruit` already enabled!

Alternatively, you can verify the `vfs_fruit` status by using the following net command while the SMB service is running:

```
net conf list |grep "vfs objects"
vfs objects = shadow_copy2 syncops fruit streams_xattr gpfs fileid time_audit
Disabling vfs_fruit support
```

ACL considerations

Normally the main file and the `._` file has the same user and group permissions. If not, then the `._` file cannot be read the fruit module does not work as expected. Thus, if the conversion does not take place as planned, then ACLs are the items to check for troubleshooting.

Remaining `._` files

Under certain circumstances some `._` files might still reappear in the file system. This can be the case if either the file is not written since the configuration change to enable `vfs_fruit` or if the data does not fit into an extended attribute due to its size, like for a large custom icon.

For more information, see *Planning for SMB* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Configuring and enabling the Swift Object protocol service

If you want to use the Cluster Export Services (CES) Swift Object service and it was not configured and enabled during the installation, configure Swift Object services.

Important:

- CES Swift Object protocol feature is not supported from IBM Storage Scale 5.2.0 onwards.
- IBM Storage Scale 5.1.8 is the last release that has CES Swift Object protocol.

- IBM Storage Scale 5.2.0 will tolerate the update of a CES node from IBM Storage Scale 5.1.8.
 - *Tolerate* means:
 - The CES node will be updated to 5.2.0.
 - Swift Object support will not be updated as part of the 5.2.0 update.
 - You may continue to use the version of Swift Object protocol that was provided in IBM Storage Scale 5.1.8 on the CES 5.2.0 node.
 - IBM will provide usage and known defect support for the version of Swift Object that was provided in IBM Storage Scale 5.1.8 until you migrate to a supported object solution that IBM Storage Scale provides.
 - CES Swift Object is replaced with IBM Storage Scale S3. For more details, refer the *S3 support overview* section the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
 - For more information about Swift Object in IBM Storage Scale, refer to the [IBM Storage Scale 5.2.0 documentation](#).
- Please contact IBM for further details and migration planning.
- 1. If a file system for the Swift Object data is created, you must create it now. For more information, see ***mmcrf command*** in *IBM Storage Scale: Command and Programming Reference Guide*.

Disabling protocol services

Exercise caution in disabling protocol services so that configuration information is not lost.

To disable a protocol service, issue the **mmces service disable** command with the appropriate service designation:

SMB

Issue the following command:

```
# mmces service disable SMB --force
```

Note: You can disable SMB only after you remove the authentication method or if the authentication method is **userdefined**.

Important: Before you disable SMB protocol services, ensure that you save all the SMB configuration information. Disabling SMB protocol services stops SMB on all CES nodes and removes all configured SMB shares and SMB settings. It removes the SMB configuration information from the CCR, removes the SMB clustered databases (trivial databases, or TBDs), and removes the SMB-related config files in the /var/mmfs/ces directory on the CES nodes.

When you re-enable SMB, you must re-create and reconfigure all the SMB settings and exports.

NFS

Issue the following command:

```
# mmces service disable NFS --force
```

Before you disable NFS protocol services, ensure that you save all the NFS configuration information by backing up the contents of the /var/mmfs/ces/nfs-config/ directory on any CES node.

Disabling the NFS service stops NFS on all CES nodes and removes the NFS configuration information from the CCR and from the /var/mmfs/ces/nfs-config/ directory. Previous exports are lost.

S3

Issue the following command:

```
# mmces service disable S3
```

Note: The S3 service disablement does not remove the S3 configuration information.

HDFS

To disable HDFS, see the *Enabling and disabling CES HDFS* section in the *IBM Storage Scale: Big Data and Analytics Guide*.

Chapter 30. Managing protocol user authentication

The system administrator can configure authentication for both object and file access either during the installation of the system or after the installation. If the authentication configuration is not configured during installation, you do it manually by using the **mmuserauth service create** command from any node in the IBM Storage Scale cluster. You can use the manual method of configuring authentication for the file and object access.

Client system authentication requirement: When you use GPFS clients or the NFS or SMB protocol to access the files in an IBM Storage Scale file system, the authentication and ID mapping of users and groups must be configured on the client operating system on which the file system or share is mounted. You must configure the appropriate directory services (AD, LDAP, or NIS) on that operating system, and users and groups must be able to log in with their user IDs and group IDs. User IDs and group IDs are the actual credentials that the file system uses to authenticate users and groups who try to access the file system through the GPFS clients.

Setting up authentication servers to configure protocol user access

Before you start configuring authentication for protocol access, ensure that the authentication server is set up and the connection between the IBM Storage Scale system and authentication server is established.

Depending on the requirement, the IBM Storage Scale system administrator needs to set up the following servers:

- Microsoft Active Directory (AD) for file and object access
- Lightweight Directory Access Protocol server for file and object access
- Keystone server to configure local, AD, or LDAP-based authentication for object access. Configuring Keystone is a mandatory requirement if you need to have object access.

AD and LDAP servers are set up externally. You can configure either an internal or external Keystone server. The installation and configuration of an external authentication server must be handled separately. The IBM Storage Scale system installation manages the installation and setup of internal Keystone server.

IBM Storage Scale system supports configuration of authentication with IPv6 address of external authentication servers.

Related concepts

[Configuring authentication and ID mapping for file access](#)

The system administrator can decide whether to configure authentication and ID mapping method either during the installation of the IBM Storage Scale system or after the installation. If the authentication configuration is not configured during installation, you can manually do it by using the **mmuserauth service create** command from any protocol node of the IBM Storage Scale system or by using the IBM Storage Scale management GUI.

[Configuring authentication for Swift Object access](#)

[Managing user-defined authentication](#)

In the user-defined mode of authentication, the user is free to select the authentication and ID-mapping methods of their choice. It is the responsibility of the administrator of the client system to manage the authentication and ID mapping for file (NFS and SMB) and object access to the IBM Storage Scale system.

[Listing the authentication configuration](#)

Use the **mmuserauth service list** command to see the authentication method that is configured in the system.

[Verifying the authentication services configured in the system](#)

Use the **mmuserauth service check** command to check whether the authentication configuration is consistent across the cluster and the required services are enabled and running. This command validates and corrects the authentication configuration files and starts any associated services if needed.

Modifying the authentication method

If data already exists or is created with the existing authentication and ID-mapping method, it is not recommended to change the authentication or the ID-mapping modes. Changing the authentication method also might invalidate the existing ACLs that are applicable to files and directories. ACLs depend on the preexisting users and group IDs.

Deleting the authentication and the ID-mapping configuration

Deleting the authentication and ID-mapping configuration results in loss of access to data. Before you remove or edit ID mappings, determine how access to data is going to be maintained.

Authentication limitations

Consider the following authentication limitations when you configure and manage the IBM Storage Scale system:

Integrating with AD server

If the authentication method is selected as AD, you must set up the AD server before you configure the authentication method in the IBM Storage Scale system.

Ensure that you have the following details before you start configuring AD-based authentication:

- IP address or hostname of the AD server. Both IPv4 and IPv6 addresses are supported.
- Domain details are as follows:
 - Domain name and realm.
 - AD admin user ID and password to join the IBM Storage Scale system as machine account into the AD domain.
- ID map role of the system is identified.
- Define the ID map range and size depending upon the maximum RID (sum of allocated and expected growth).
- Primary DNS is added in the `/etc/resolv.conf` file on all the protocol nodes. It resolves the authentication server system with which the IBM Storage Scale system is configured. This primary DNS addition is a mandatory requirement when AD is used as the authentication server. Because the DNS must be able to resolve the host domain and its trusted domains of interest. The manual changes done to the configuration files might get overwritten by the Operating System's network manager. So, ensure that the DNS configuration is persistent even after you restart the system. For more information about the circumstances where the configuration files are overwritten, see the corresponding operating system documentation.
- During the AD join process, a computer account that has the same name as the NetBIOS name is searched within the AD domain that will be joined. If the name is not found, a new computer entry is created in the standard location (CN=Computers). If the user chooses to pre-create computer accounts for IBM Storage Scale in the AD domain within a particular organizational unit, the computer account must be created with a valid name and it must be passed as the NetBIOS name while configuring the IBM Storage Scale system. After the account is created on the AD server, the system must be joined to the AD domain.

To achieve high-availability, you can configure multiple AD domain controllers. While you configure the AD-based authentication, you do not need to specify multiple AD servers in the command line to achieve high-availability. The IBM Storage Scale system queries the specified AD server for relevant details and configures itself for the AD-based authentication. The IBM Storage Scale system relies on the DNS server to identify the set of available AD servers that are currently available in the environment that is serving the same domain system.

Related concepts

[Integrating with LDAP server](#)

If LDAP-based authentication is selected, ensure that the LDAP server is set up with the required schemas to handle the authentication and ID-mapping requests. If you need to support SMB data access, LDAP schema must be extended before you configure the authentication.

Integrating with LDAP server

If LDAP-based authentication is selected, ensure that the LDAP server is set up with the required schemas to handle the authentication and ID-mapping requests. If you need to support SMB data access, LDAP schema must be extended before you configure the authentication.

Ensure that you have the following details before you start configuring LDAP-based authentication:

- Default values are used unless you specify domain details such as **--base-dn** as prefixes of groups and users. The default user group suffix is **--base-dn** and the default user suffix is **--base-dn**.
- IP address or hostname of LDAP server. Both IPv4 and IPv6 addresses are supported.
- Admin user ID and password of LDAP server that is used during LDAP simple bind and for LDAP searches.
- The secret key that you provided for encrypting or decrypting passwords unless you disabled prompting for the key.
- NetBIOS name that is to be assigned for the IBM Storage Scale system.
- If you need secure communication between the IBM Storage Scale system and LDAP, the CA signed certificate that is used by the LDAP server for TLS communication must be placed at the specified location in the system.
- If you are using LDAP with Kerberos, create a Kerberos keytab file by using the MIT KDC infrastructure.
- Primary DNS is added in the `/etc/resolv.conf` file on all the protocol nodes. It resolves the authentication server system with which the IBM Storage Scale system is configured. The manual changes done to the configuration files might get overwritten by the operating system's network manager. So, ensure that the DNS configuration is persistent even after you restart the system. For more information about the circumstances where the configuration files are overwritten, see the corresponding operating system documentation.

Related concepts

[Integrating with AD server](#)

If the authentication method is selected as AD, you must set up the AD server before you configure the authentication method in the IBM Storage Scale system.

Setting up LDAP server prerequisites

Before you start configuring the IBM Storage Scale system with LDAP server, the following external LDAP server prerequisites must be met:

- The LDAP server must already be configured.
- Enable TLS encryption on the LDAP server, if you need to secure communication between the IBM Storage Scale system and LDAP server. Details on configuring SSL or TLS encryption on the server can be obtained from the *OpenLDAP Administrator's Guide*.
- To access SMB shares, LDAP user information must be updated with unique Samba attributes in addition to the attributes that are stored for a normal LDAP user. Ensure that these required Samba attributes are present in the LDAP user entries.
- Ensure you do not have the same user name for different organizational units of the LDAP server that is configured with the IBM Storage Scale system.

LDAP bind user requirements

When an IBM Storage Scale system is configured with LDAP as the authentication method, the IBM Storage Scale system needs to connect to the LDAP server by using an administrative user ID and password. This administrative user is referred as bind user.

It is recommended that the bind user is given enough privileges that are required by the storage system to mitigate any security concerns.

This bind user must at least have permission to query users and groups that are defined in the LDAP server to allow storage system to authenticate these users. The bind user information (bind_dn) is also used by the Samba server while making LDAP queries to retrieve required information from the LDAP server.

Note: In the following sections, it is assumed that the user account for the bind user exists in the LDAP directory server. The bind user distinguished name (also known as dn) used in the following examples is uid=ibmbinduser,ou=people,dc=ldapserver,dc=com. This name needs to be updated based on the bind user that is used with the IBM Storage Scale system.

For data access method file and authentication type LDAP, the bind user and password are not required when you issue the **mmuserauth** create command with the **--enable-anonymous-bind** parameter.

Important: This authentication type works if the LDAP server supports anonymous binding. This authentication type is valid for NFS exports only.

OpenLDAP server ACLs

The OpenLDAP server ACLs define the privileges that are required for the bind user.

The following example uses ACLs that are required for the bind user and other type of users for the sake of completeness. It is likely that a corporate directory server has those ACLs that are configured already and only the entries for the bind user need to be merged correctly in the slapd configuration file. This file is generally, in the /etc/openldap/slapd.conf file on Linux systems. Follow the ACL ordering rules to ensure that correct ACLs are applied.

```
### some attributes need to be readable so that commands like 'id user',
'getent' etc can answer correctly.
access to attrs=cn,objectClass,entry,homeDirectory,uid,uidNumber,
gidNumber,memberUid
by dn="uid=ibmbinduser,ou=people,dc=ldapserver,dc=com" read
```

```
###The following will not list userPassword when ldapsearch is
performed with bind user.
```

```
### Anonymous is needed to allow bind to succeed and users to
authenticate, should be
a pre-existing entry already.
access to attrs=userPassword
by dn="uid=ibmbinduser,ou=people,dc=ldapserver,dc=com" auth
by self write
by anonymous auth
by * none
```

```
### Storage system needs to be able to find samba domain account
specified on the mmuserauth service create command.
```

```
###It is strongly recommended that domain account is pre-created
to ensure
```

```
###consistent access to multiple storage systems.
```

```
###Uncomment ONLY if you want storage systems to create domain
account when it does not exist.
#access to dn.base="dc=ldapserver,dc=com"
#      by dn="uid=ibmbinduser,ou=people,dc=ldapserver,dc=com" write
#      by * none

access to dn.regex="sambadomainname=[^,]+,dc=ldapserver,dc=com"
by dn="uid=ibmbinduser,ou=people,dc=ldapserver,dc=com" read
```

```

by * none

##### all samba attributes need to be readable for samba access
access to attrs=cn,sambaLMPassword,sambaNTPassword,sambaPwdLastSet,
sambaLogonTime,sambaLogoffTime,sambaKickoffTime,sambaPwdCanChange,
sambaPwdMustChange,sambaAcctFlags,displayName,sambaHomePath,
sambaHomeDrive,sambaLogonScript,sambaProfilePath,description,
sambaUserWorkstations,sambaPrimaryGroupSID,sambaDomainName,
sambaMungedDial,sambaBadPasswordCount,sambaBadPasswordTime,
sambaPasswordHistory,sambaLogonHours,sambaSID,sambaSIDList,
sambaTrustFlags,sambaGroupType,sambaNextRid,sambaNextGroupRid,
sambaNextUserRid,sambaAlgorithmicRidBase,sambaShareName,
sambaOptionName,sambaBoolOption,sambaIntegerOption,
sambaStringOption,sambaStringListoption
    by dn="uid=ibmbinduser,ou=people,dc=ldapserver,dc=com" read
    by self read
    by * none

##### Need write access to record bad failed login attempt
access to attrs=cn,sambaBadPasswordCount,sambaBadPasswordTime,
sambaAcctFlags by dn="uid=ibmbinduser,ou=people,dc=ldapserver,
dc=com" write

##### Required to check samba schema
access to dn.base=* by dn="uid=ibmbinduser,ou=people,
dc=ldapserver,dc=com" read

```

IBM Storage Protect Directory Server ACLs

The IBM Storage Protect Directory Server ACLs define the privileges that are required for the bind user, when the user uses IBM Storage Protect Directory Server.

These ACLs are provided in the LDIF format and can be applied by submitting the **ldapmodify** command.

```

dn: dc=ldapserver,dc=com
changetype: modify

add: ibm-filterAclEntry

ibm-filterAclEntry:access-id:uid=ibmbinduser,ou=people,dc=ldapserver,dc=com:
(objectClass=sambaSamAccount):normal:rsc:sensitive:rsc:critical:rsc
-
add:ibm-filterAclEntry

ibm-filterAclEntry:access-id:uid=ibmbinduser,ou=people,dc=ldapserver,dc=com:
(objectclass=sambaDomain):normal:rwsc:sensitive:rwsc:critical:rwsc

dn:uid=ibmbinduser,ou=people,dc=ldapserver,dc=com

add:aclEntry

aclentry: access-id:uid=ibmbinduser,ou=people,dc=ldapserver,dc=com:at.cn:r:at.
objectClass:r:at.homeDirectory:r:at.uid:r:at.uidNumber:s:

at.gidNumber:r:at.memberUid:r:at.userPassword:sc:at.sambaLMPassword:r:at.
sambaNTPassword:r:at.sambaPwdLastSet:r:at.sambaLogonTime:r:

at.sambaLogoffTime:r:at.sambaKickoffTime:r:at.sambaPwdCanChange:r:at.
sambaPwdMustChange:r:at.sambaAcctFlags:r:at.displayName:r:

at.sambaHomePath:r:at.sambaHomeDrive:r:at.sambaLogonScript:r:at.sambaProfilePath:
r:at.description:r:at.sambaUserWorkstations:r:

at.sambaPrimaryGroupSID:r:at.sambaDomainName:r:at.sambaMungedDial:r:at.
sambaBadPasswordCount:r:at.sambaBadPasswordTime:r:
at.sambaPasswordHistory:r:at.sambaLogonHours:r:at.sambaSID:r:at.sambaSIDList:r:at.
sambaTrustFlags:r:at.sambaGroupType:r:
at.sambaNextRid:r:at.sambaNextGroupRid:r:at.sambaNextUserRid:r:at.
sambaAlgorithmicRidBase:r:at.sambaShareName:r:at.sambaOptionName:r:

at.sambaBoolOption:r:at.sambaIntegerOption:r:at.sambaStringOption:r:at.

```

```
sambaStringListoption:r:at.sambaBadPasswordCount:rwsc:  
at.sambaBadPasswordTime:rwsc:at.sambaAcctFlags:rwsc  
  
#### Storage system needs to be able to find samba domain account specified  
on the mmuserauth service create command.  
  
####It is strongly recommended that domain account is pre-created to ensure  
####consistent access to multiple storage systems.  
  
####Uncomment ONLY if you want storage systems to create domain account when  
it does not exist.  
  
dn: dc=ldapserver,dc=com  
  
changetype: modify  
  
add:ibm-filterAclEntry  
  
ibm-filterAclEntry:access-id:uid=ibmbinduser,ou=people,dc=ldapserver,  
dc=com:(objectclass=domain):object:grant:a
```

See *IBM Tivoli Directory Server Administration Guide* for information about applying these ACLs on the IBM Storage Protect Directory Server.

Updating LDAP user information with Samba attributes

If you need to support SMB data access, LDAP schema must be extended to store more attributes such as SID, Windows password hash to the POSIX user object. To use Samba accounts, update LDAP user information with unique Samba attributes.

The following sample LDIF file shows the minimum required samba attributes:

Note: Attributes must be separated with a dash as the first and only character on a separate line.

Perform the following steps to create the values for `sambaNTPassword`, `sambaPwdLastSet`, and `SambaAcctFlags`, which must be generated from a PERL module:

1. Download the module from <http://search.cpan.org/~bjkuit/Crypt-SmbHash-0.12/SmbHash.pm>. Create and install the module by following the readme file.
 2. Use the following PERL script to generate the LM and NT password hashes:

```
# cat /tmp/Crypt-SmbHash-0.12/gen_hash.pl
#!/usr/local/bin/perl
use Crypt::SmbHash;
$username = $ARGV[0];
$password = $ARGV[1];
if ( !$password ) {
    print "Not enough arguments\n";
    print "Usage: $0 username password\n";
    exit 1;
```

```

}
$uid = (getpwnam($username))[2];
my ($login, undef, $uid) = getpwnam($ARGV[0]);
ntlmgen $password, $lm, $nt;
printf "%s:%d:%s:%s:[%-11s]:LCT-%08X\n", $login, $uid, $lm, $nt, "U", time;

```

3. Generate the password hashes for any user as in the following example for the user test01:

```

# perl gen_hash.pl SMBUser test01
:0:47F9DBCCD37D6B40AAD3B435B51404EE:82E6D500C194BA5B9716495691FB7DD6:
[U] :LCT-4C18B9FC

```

Note: The output contains login name, uid, LM hash, NT hash, flags, and time, with each field separated from the next by a colon. The login name and uid are omitted because the command was not run on the LDAP server.

4. Use the information from step 3 to update the LDIF file in the format that is provided in the example at the beginning of this topic.

- To generate the sambaPwdLastSet value, use the hexadecimal time value from step 3 after the dash character and convert it into decimal.
- A valid samba SID is required for a user to enable that user's access to an IBM Storage Scale share. To generate the samba SID, multiply the user's UID by 2 and add 1000. The user's SID must contain the samba SID from the sambaDomainName, which is either generated or picked up from the LDAP server, if it exists. The following attributes for sambaDomainName LDIF entry are required:

```

dn: sambaDomainName=(IBM Spectrum Scale system),dc=ibm,dc=com
sambaDomainName: (IBM Spectrum Scale system name)
sambaSID: S-1-5-21-1528920847-3529959213-2931869277
sambaPwdHistoryLength: 0
sambaMaxPwdAge: -1
sambaMinPwdAge: 0

```

This entry can be created by the LDAP server administrator by using either of the following two methods:

- Write and run a bash script similar to the following example:

```

sambaSID=
for num in 1 2 3 ;do
    randNum=$(od -vAn -N4 -tu4 < /dev/urandom | sed -e 's/ //g')
    if [ -z "$sambaSID" ];then
        sambaSID="S-1-5-21-$randNum"
    else
        sambaSID="${sambaSID}-$randNum"
    fi
done
echo $sambaSID

```

Then, use the samba SID that is generated to create the LDIF file. The sambaDomainName must match the IBM Storage Scale system name.

- When you run the **mmuserauth service create** command, the system creates the sambaDomainName, if it does not exist.

The sambaSID for every user must have the following format: (samba SID for the domain)-(userID*2+1000). For example, S-1-5-21-1528920847-3529959213-2931869277-1102

Note: To enable access by using the same LDAP server domain to more than one IBM Storage Scale system or another IBM NAS like IBM SONAS or IBM V7000 Unified, the Samba domain SID prefix of all of the systems must match. The Samba domain SID prefix is used to prepare the SID of users or groups that are planning to access the IBM Storage Scale system by using CIFS. So, if you change the Samba domain SID for an IBM Storage Scale system on the LDAP server, you must restart the CES Samba service on that IBM Storage Scale system for the change to take effect.

5. Submit the **ldapmodify** command as shown in the following example to update the user's information:

```
# ldapmodify -h localhost -D cn=Manager,dc=ibm,dc=com -W -x -f /tmp/samba_user.ldif
```

Configuring authentication and ID mapping for file access

The system administrator can decide whether to configure authentication and ID mapping method either during the installation of the IBM Storage Scale system or after the installation. If the authentication configuration is not configured during installation, you can manually do it by using the **mmuserauth service create** command from any protocol node of the IBM Storage Scale system or by using the IBM Storage Scale management GUI.

Related concepts

[Setting up authentication servers to configure protocol user access](#)

Before you start configuring authentication for protocol access, ensure that the authentication server is set up and the connection between the IBM Storage Scale system and authentication server is established.

[Configuring authentication for Swift Object access](#)

[Managing user-defined authentication](#)

In the user-defined mode of authentication, the user is free to select the authentication and ID-mapping methods of their choice. It is the responsibility of the administrator of the client system to manage the authentication and ID mapping for file (NFS and SMB) and object access to the IBM Storage Scale system.

[Listing the authentication configuration](#)

Use the **mmuserauth service list** command to see the authentication method that is configured in the system.

[Verifying the authentication services configured in the system](#)

Use the **mmuserauth service check** command to check whether the authentication configuration is consistent across the cluster and the required services are enabled and running. This command validates and corrects the authentication configuration files and starts any associated services if needed.

[Modifying the authentication method](#)

If data already exists or is created with the existing authentication and ID-mapping method, it is not recommended to change the authentication or the ID-mapping modes. Changing the authentication method also might invalidate the existing ACLs that are applicable to files and directories. ACLs depend on the preexisting users and group IDs.

[Deleting the authentication and the ID-mapping configuration](#)

Deleting the authentication and ID-mapping configuration results in loss of access to data. Before you remove or edit ID mappings, determine how access to data is going to be maintained.

[Authentication limitations](#)

Consider the following authentication limitations when you configure and manage the IBM Storage Scale system:

Prerequisites

Ensure that the following requirements are met before you start configuring an authentication method for file access.

Related concepts

[Configuring file authentication by using CLI](#)

You need to use the **mmuserauth service create** command to configure user authentication by using CLI.

Related tasks

[Configuring file authentication by using GUI](#)

You can configure an authentication method or view the existing authentication method that is used for Network File System (NFS) and Server Message Block (SMB) users from the **Services > File Authentication** page of the GUI.

Prerequisite for configuring Kerberos-based SMB access

The following requirements must be met to configure IBM Storage Scale for Kerberized SMB access:

- The time must be synchronized across the KDC server, the IBM Storage Scale cluster protocol nodes, and the SMB clients, or else access to an SMB share could be denied.
- In MIT KDC configurations for the SMB services, the service principal name must use the NetBIOS name and the realm name. For example, if the NetBIOS name is FOO and the realm is KDC.COM, the service principal name should be cifs/foo@KDC.COM. The NetBIOS name is the value specified for the option --netbios_name in the **mmuserauth** command. The realm may be discovered from the value stored for Alt_Name returned from the command: wbinfo -D <domain>.
- The clients should use only the NetBIOS name when accessing an SMB share. Using any other name or IP address might either cause a failure to connect or fallback to NTLM authentication.
- With Active Directory KDC, you can use DNS alias (CNAME) for Kerberized SMB access. To use the alias, you must register the DNS alias (CNAME) record for the NetBIOS name (system account name) using the SetSPN tool available on Active Directory server. For example, if the NetBIOS name is FOO and the DNS alias is BAR, use the SetSPN tool from the command prompt of the Active Directory server to register the record, "setspn -A cifs/BAR FOO". Not registering the DNS alias record for the NetBIOS name might cause access to the SMB shares to be denied with the error code, KDC_ERR_S_SPRINCIPAL_UNKNOWN.
- On Linux clients, to use Kerberized SMB access for IBM Storage Scale configured with MIT KDC, you must at least have the 3.5.9 version of Samba client installed. The Linux clients having an older Samba client might encounter the following error, while trying to access SMB shares:

```
ads_krb5_mk_req: krb5_get_credentials failed for foo$@KDC.COM (Server not found in Kerberos database)
cli_session_setup_kerberos: spnego_gen_negTokenTarg failed: Server not found in Kerberos database
```

To determine if a client has authenticated via Kerberos, either verify at the client or collect a protocol trace:

```
mmptrace start smb -c x.x.x.x
```

Replace x.x.x.x with the IP address of the client system access IBM Storage Scale to be verified.

Access the IBM Storage Scale SMB service from that client.

Then, issue the command:

```
mmptrace stop smb
```

Extract the compressed trace files and look for the file ending with smbd.log. If that file contains an entry similar to "Kerberos ticket principal name is..." then Kerberos is being used.

Note: It is not recommended to run for extended periods of time at log levels higher than 1 as this could impact performance.

Authentication considerations for NFSv4 based access

Authentication considerations for NFSv4 based access are as follows.

NFSv4 username-mapping configuration

1. To enable NFSv4 access, the NFS server username-mapping configuration on IBM Storage Scale must be updated.

```
# mnfs config change "IDMAPD_DOMAIN=myDomain.com"
```

2. On an NFS client, ID map configuration must also be updated to reflect the same domain name as defined on an NFS server. Additionally ID-mapping service must be started.

For example, on RHEL 7.x NFS clients:

- The ID map configuration file name is `/etc/idmapd.conf`. Update the `domain` attribute in the file to reflect the domain name that is defined on the NFS server.
- Start `nfs.idmap` service.

Note: The ID map configuration file and the ID-mapping service can be different on various OS platforms.

Prerequisites for configuring Kerberos based NFS access

Certain requirements must be met to configure IBM Storage Scale for Kerberized NFS access.

General requirements

- For Kerberized NFS access, time must be synchronized across the KDC server, the IBM Storage Scale cluster protocol nodes, and the NFS clients. Otherwise, access to an NFS export might be denied.
- For Kerberized NFSv3 access, NFS clients must mount NFS exports by using one of the configured CES IP addresses.
- For Kerberized NFSv4 access, NFS clients can mount NFS exports by using either "one of the configured CES IP addresses" or the "system account name" that is configured for FILE protocols authentication. The "system account name" is the value that is specified for the `--netbios-name` option in the **mmuserauth CLI** command during FILE protocols authentication configuration.

IBM Storage Scale NFS server configuration for Kerberos access

- To enable NFS Kerberos access, update the NFS server configuration with the Kerberos realm name. Issue the following command to configure NFS configuration parameter LOCAL_REALMS:

```
# mmnfs config change "LOCAL_REALMS=MYREALM.COM"
```

Set this attribute to the KDC REALM value.

Note: Specify the realm name in capital letters.

- Configure the same local realms value (for example, MYREALM.COM here) on all NFS Kerberos clients (for example, on RHEL NFS clients set Local-Realms attribute in the `/etc/idmapd.conf` file). This configuration file might be different on various client OS systems.
- On NFS client, ID map configuration must also be updated to reflect the same realm name as defined on NFS server. Additionally, the service for establishing Kerberos access with NFS server must also be started. For example, on RHEL 7.X NFS clients, the ID map configuration file name is `/etc/idmapd.conf`. Update the Local-Realms attribute in the file to reflect the Kerberos realm that is defined on NFS server and then start the `nfs.secure` service.

Note: The ID map configuration file and service to establish secure access can differ on various OS platforms.

Considerations for LDAP-based authentication schemes:

- In LDAP-based authentication schemes, administrators must generate keytab file before the FILE protocols authentication configuration. The keytab file must be generated on the KDC server and then copied to path `/var/mmfs/tmp/` on the IBM Storage Scale node. The **mmuserauth** command must be initiated from the node where the keytab file is copied.
- The keytab file must contain NFS service principals of short name and FQDN of the "system account name". The service principal name format is `nfs/<system account name>@<KERBEROS REALM>`. For example, if the "system account name" is FOO, "system account FQDN" is FOO.MYDOMAIN.COM and the "realm" is MYREALM.COM, then service principals that are required to be created must be `nfs/FOO@MYREALM.COM` and `nfs/FOO.MYDOMAIN.COM@MYREALM.COM`.

- The realm name is the value that is specified for the --kerberos-realm-option in the **mmuserauth** command.

Considerations for AD-based authentication schemes:

- In Active Directory based authentication schemes, administrators need not prepare a keytab file. The **mmuserauth CLI** command prepares keytab file during FILE protocols authentication configuration. It adds NFS service principals of short name and FQDN for "system account name" in the local keytab file that is placed at /var/mmfs/etc/krb5_scale.keytab on all the protocol nodes in the CES cluster.
- User must specify the --enable-nfs-kerberos option in the **mmuserauth** command to activate the NFS Kerberized access to IBM Storage Scale.

Configuring file authentication by using CLI

You need to use the **mmuserauth service create** command to configure user authentication by using CLI.

You can configure the following external authentication servers for file access:

- Active Directory (AD)
- Lightweight Directory Access Protocol (LDAP)
- Network Information Service (NIS)

Before you configure the authentication method, ensure that the following RPMs are installed on all the protocol nodes before you start configuring the authentication method:

Note: If you try to configure the file authentication method manually, with the **mmuserauth** command, the command displays an error message if the required RPMs are not installed on the nodes. The error output includes a list of nodes in which some RPMs are not installed and a list of the missing RPMs for each node.

On Red Hat Enterprise Linux nodes

- For AD:**
 - bind-utils
 - krb5-workstation
- For LDAP:**
 - openldap-clients
 - sssd and its dependencies (particularly sssd-common, sssd-ldap, and sssd-tools). It is a good idea to install all the dependencies, as in the following example:

```
yum install sssd*
```

 - krb5-workstation only if Kerberized authentication is planned.
- For NIS:**
 - sssd and its dependencies (particularly sssd-common and sssd-proxy)
 - ypbind and its dependencies (yp-tools)

On SLES nodes

- For AD:**
 - bind-utils
 - krb5-client
- For LDAP:**
 - openldap2-client

- sssd and its dependencies (particularly sssd-common, sssd-ldap, sssd-krb5, and sssd-tools). It is a good idea to install all the dependencies, as in the following example:

```
zypper install sssd*
```

- krb5-client only if Kerberized authentication is planned.

- **For NIS:**

- sssd and its dependencies (particularly sssd-common and sssd-proxy)
- ypbind and its dependencies (yp-tools)

On Ubuntu nodes

- **For AD:**

- dnsutils
- krb5-user (only if Kerberos authentication is planned)

- **For LDAP:**

- ldap-utils
- krb5-user (only if Kerberos authentication is planned)
- sssd and its dependencies (particularly sssd-tools). It is a good idea to install all the dependencies, as in the following example:

```
apt-get install sssd
```

- **For NIS:**

- sssd and its dependencies (particularly sssd-common and sssd-proxy)
- nis and libslp1 (nis package automatically installs the libslp1 package)

Related concepts

Prerequisites

Ensure that the following requirements are met before you start configuring an authentication method for file access.

Related tasks

[Configuring file authentication by using GUI](#)

You can configure an authentication method or view the existing authentication method that is used for Network File System (NFS) and Server Message Block (SMB) users from the **Services > File Authentication** page of the GUI.

Configuring AD-based authentication for file access

You can configure Microsoft Active Directory (AD) as the authentication server to manage the authentication requests and to store user credentials.

You can configure AD-based authentication with the following ID mapping methods:

- RFC2307
- Automatic
- LDAP

RFC2307 ID mapping

In the RFC2307 ID mapping method, the user and group IDs are stored and managed in the AD server and these IDs are used by the IBM Storage Scale system during file access. The RFC2307 ID mapping method is used when you want to have multiprotocol access. That is, you can have both NFS and SMB access over the same data.

Automatic ID mapping

In the automatic ID mapping method, user ID and group ID are automatically generated and stored within the IBM Storage Scale system. When an external ID mapping server is not present in the environment or cannot be used, then this ID mapping method can be used. This method is typically used if you have SMB only access and do not plan to deploy multiprotocol access. That is, the AD-based authentication with automatic ID mapping is not used if you need to allow NFS and SMB access to the same data.

LDAP ID mapping

In the LDAP mapping method, user ID and group ID are stored and managed in the LDAP server, and these IDs are used by the IBM Storage Scale system during file access. The LDAP ID mapping method is used when you want to have multiprotocol access. That is, you can have both NFS and SMB access over the same data.

Setting up a range of ID maps that can be allotted to the users

You can optionally specify the pool of values from which the UIDs and GIDs are assigned by the IBM Storage Scale system to Active Directory (AD) users and groups. When a user or group is defined in Active Directory, it is identified by a security identifier (SID), which includes a component that is called Relative Identifier (RID). The RID value depends on the number of users and groups in the AD domain. The **--idmap-range** and **--idmap-range-size** parameters of the **mmuserauth service create** command specify the pool from which UIDs and GIDs are assigned by the IBM Storage Scale system to AD users and group of users.

The ID map range is defined between a minimum and maximum value. The default minimum value is 10000000 and the default maximum value is 299999999, and the default range size is 1000000. This allows for a maximum of 290 unique AD domains.

The ID map range size specifies the total number of UIDs and GIDs that are assignable per domain. For example, if range is defined as 10000-20000, and range size is defined as 2000 (**--idmap-range 10000-20000 : 2000**), five domains can be mapped, each consisting of 2000 IDs. Make sure when you define the range size value it is defined so that at least three domains can be mapped. The range size is identical for all AD domains that are configured by the IBM Storage Scale system. Choose an ID map range size that allows for the highest anticipated RID value among all of the anticipated AD users and group of users in all of the anticipated AD domains. Make sure that you define the range size value to take into account the planned growth in the number of AD users and groups of users. The ID map range size cannot be changed after the IBM Storage Scale system is configured with AD as the authentication server.

Whenever a user or user group from an AD domain accesses the IBM Storage Scale system, a range is allocated per domain. UID or GID for a user or user group is allocated depending upon this range and the RID of the user or user group. If RID of any user or group is greater than the range size, then that user or user group is mapped into extension ranges depending upon the number of available ranges. If the number of ranges (default value is 290) runs out, then mapping requests for a new user or user group (or new extension ranges for user and group that is already known) are ignored and thus that user and user group cannot access the data.

Choosing range size

1. Determine the highest AD RID that is currently assigned. You can use the **dcdiag** command at the command prompt of the operating system of the server that is hosting AD to determine the value of the **rIDNextRID** attribute. For example:

```
# dcdiag /s:IP_of_system_hosting_AD /v /test:ridmanager
```

Specifically,

```
C:\Program Files\Support Tools>dcdiag /s:10.0.0.123 /v /test:ridmanager
```

The following output is displayed:

```
Starting test: RidManager
  * Available RID Pool for the Domain is 1600 to 1073741823
  * win2k8.pollux.com is the RID Master
  * DsBind with RID Master was successful RFC23071
  * rIDAllocationPool is 1100 to 1599
  * rIDPreviousAllocationPool is 1100 to 1599
  * rIDNextRID: 1174
```

In this example, the `rIDNextRID` value is 1174. Another way to determine the current value for `rIDNextRID` is to run an LDAP query on the following DN Path:

```
CN=Rid Set,Cn=computername,ou=domain controllers,DC=domain,DC=COM
```

If there is more than one domain controller serving the AD domain, determine the highest RID among all of the domain controllers. Similarly, if there is more than one domain, determine the highest RID among all of the domains.

2. Estimate the expected number of users and groups that might be added in future, in addition to the current number of users and groups.
3. Add the highest RID determined in step 1 to the number of users and groups that were estimated in the previous step. The result is the estimate for the value of the range size.

Considerations for changing the ID map range and range size

If the IBM Storage Scale system is configured to use AD-based authentication, only the maximum value of ID map range can be changed. All other changes to ID map range and size, except increasing the maximum value of ID map range require reconfiguration of authentication, which results in loss of access to data. For example, if you used the `--idmap-range` as 3000-10000 and `--idmap-range-size` as 2000, you can increase only the value 10000 to accommodate more users per domain, without having an impact on the access to the data.

To change the ID mapping of an existing AD-based authentication configuration, either to change the range minimum value, decrease the range maximum value, or change the range size, you must complete the following steps:

Note: The `mmuserauth service remove` command results in loss of access.

1. Submit the `mmuserauth service remove` command and do not specify the `--idmapdelete` option.
2. Submit the `mmuserauth service remove` command and do specify the `--idmapdelete` option.
3. Submit the `mmuserauth service create` command with the options and values that you want for the new Active Directory configuration.

Important: If you do not perform the preceding three steps in sequence, results are unpredictable and can include complete loss of data access.

Prerequisite for configuring AD-based authentication for file access

See “[Integrating with AD server](#)” on page 294 for more information on the prerequisites for integrating AD server with the IBM Storage Scale system.

You must run the `mmuserauth service create` command with the following parameters to create AD-based authentication for file access:

- `--type ad`
- `--data-access-method file`
- `--servers server host name or IP address`
- `--netbios-name netBiosName`
- `--user-name admin-username`
- `--unixmap-domains <unixDomainMap>`. This option is mandatory if RFC2307 ID mapping is used.
For example, `--unixmap-domains DOMAINS(5000-20000)`. Specifies the Active Directory domains

for which user ID and group ID must be fetched from the Active directory server (RFC2307 schema attributes).

- `--idmap-role master | subordinate`. While you use the automatic ID mapping for the same ID maps on systems that share Active File Manager (AFM) relationship, you must export the ID mappings from the system whose ID map role is master to the system whose ID map role is subordinate.

For more information about each parameter, see the *mmuserauth service create* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Prerequisites for configuring AD with RFC2307

The following prerequisites are specific to AD with RFC2307 configuration:

- RFC2307 schema is extended on the AD and all UNIX attributes (including UID and GID) are populated.
- If a trusted domain is configured with ID mapping from RFC2307, the trusted domain must have two-way trust with the host domain. This host domain is the Active Directory domain that is configured for use with the IBM Storage Scale system. For example, assume that trusted relationships are X, Y, Z, and the IBM Storage Scale system is configured with domain X as the host domain. If RFC2307 ID mappings are required for domains Y and Z, domains Y and Z must each have a two-way trust with the domain X. X <-> Y ; X <-> Z.
- User and group in the Active Directory domain, which is configured with ID mapping from RFC2307, must have a valid UID and a valid GID assigned to enable access to IBM Storage Scale system exports. The UID and GID number that is assigned must be within the ID map range that is specified in the **mmuserauth service create** command. Any users or groups from this domain that do not have UID or GID attributes configured are denied access.

Note: The primary Windows group that is assigned to an AD user must have a valid GID assigned within the specified ID-mapping range. The primary Windows group is usually located in the Member Of tab in the user's properties. The primary Windows group is different from the UNIX primary group, which is listed in the UNIX Attributes tab. A user is denied access if a valid GID is not assigned to the user's Windows primary group. The UNIX primary group attribute is ignored.

In a case of a mutual trust setup between two independent AD domains, DNS forwarding must be configured between the two trusts.

Configuring AD-based authentication with automatic ID mapping

When the IBM Storage Scale system is configured for AD-based authentication, automatic ID mapping method can be used to create UID or GID of a user or group respectively. The ID maps are stored within the IBM Storage Scale system.

The following provides an example of how to configure an IBM Storage Scale system with Active Directory and automatic ID mapping.

1. Issue the **mmuserauth service create** command as shown in the following example:

```
# mmuserauth service create --type ad --data-access-method file --netbios-name  
ess --user-name administrator --idmap-role master --servers myADserver --idmap-range-size  
1000000  
--idmap-range 10000000-299999999
```

The system displays the following output:

```
File authentication configuration completed successfully.
```

2. Verify the authentication configuration by issuing the **mmuserauth service list** command as shown in the following example:

```
# mmuserauth service list
```

The system displays the following output:

```
FILE access configuration : AD  
PARAMETERS          VALUES
```

```

ENABLE_NFS_KERBEROS      false
SERVERS                  "*"
USER_NAME                ess$
NETBIOS_NAME              ess
IDMAP_ROLE                master
IDMAP_RANGE               10000000-299999999
IDMAP_RANGE_SIZE          1000000
UNIXMAP_DOMAINS           none
LDAPMAP_DOMAINS           none

OBJECT access not configured
PARAMETERS                 VALUES
-----
```

3. Verify the user resolution on the system:

```
# id "DOMAIN\user1"
uid=12001172(DOMAIN\user1) gid=12001174(DOMAIN\group1) groups=12001174
(DOMAIN\group1),12001172(DOMAIN\user1),12000513(DOMAIN\domain users),
11000545(BUILTIN\users)
```

4. To configure an IBM Storage Scale system with Active Directory that has IPv6 address, issue the following command:

```
# mmuserauth service create --type ad --data-access-method file --servers
[2001:192::e61f:122:feb7:5df0]
--netbios-name specscale --user-name adUser --idmap-role master --idmap-range-size
1000000 --idmap-range 10000000-299999999
```

The system displays the following output:

```
File authentication configuration completed successfully.
```

5. To verify the authentication configuration with Active Directory that has IPv6 address, issue the **mmuserauth service list** command as shown in the following example:

```
# mmuserauth service list
```

The system displays the following output:

```

FILE access configuration : AD
PARAMETERS                 VALUES
-----
ENABLE_NFS_KERBEROS      false
SERVERS                  "*"
USER_NAME                adUser$
NETBIOS_NAME              specscale
IDMAP_ROLE                master
IDMAP_RANGE               10000000-299999999
IDMAP_RANGE_SIZE          1000000
UNIXMAP_DOMAINS           none
LDAPMAP_DOMAINS           none

OBJECT access not configured
PARAMETERS                 VALUES
-----
```

Configuring AD-based authentication with RFC2307 ID mapping

When the IBM Storage Scale is configured for the AD-based authentication with the RFC2307 ID mapping method, ID mappings are read from the AD server. The value that is stored in the uidNumber attribute for a user and the gidNumber attribute for a group is read from the AD server.

This ID mapping method is useful when:

- You have user IDs and group IDs that are populated on the AD server.
- You want to host data on IBM Storage Scale system that NFS and SMB clients access.
- You want to host multiple IBM Storage Scale systems in an AFM relationship.

If you use an AD-based authentication and the ID maps are not configured with RFC2307, the IBM Storage Scale system uses the automatic ID mappings by default. In multiple AD-domain setups, IBM

Storage Scale system reads the IDs from the AD server for the AD domains that are configured with RFC2307 ID mapping. The remaining AD domains are configured with the automatic ID mapping mode.

Configuring Active Directory with RFC2307 mapping

The following steps provide an example of configuring Active Directory (AD) with RFC2307 mapping

1. Issue the **mmuserauth service create** command as shown in the following example:

```
# mmuserauth service create --type ad --data-access-method file --netbios-name  
ess --user-name administrator --idmap-role master --servers myADserver  
--idmap-range-size 1000000 --idmap-range 10000000-299999999  
--unixmap-domains 'DOMAIN(5000-20000)'
```

The system displays the following output:

```
File authentication configuration completed successfully.
```

2. Issue the **mmuserauth service list** to verify the authentication configuration as shown in the following example:

```
# mmuserauth service list
```

The system displays the following output:

```
FILE access configuration : AD  
PARAMETERS          VALUES  
-----  
ENABLE_NFS_KERBEROS    false  
SERVERS                "*"  
USER_NAME               ess$  
NETBIOS_NAME            ess  
IDMAP_ROLE              master  
IDMAP_RANGE             10000000-299999999  
IDMAP_RANGE_SIZE        1000000  
UNIXMAP_DOMAINS         DOMAIN(5000-20000:win)  
LDAPMAP_DOMAINS         none  
  
OBJECT access not configured  
PARAMETERS          VALUES
```

3. Verify the user name resolution on the system. Confirm that the resolution is showing IDs that are pulled from RFC2307 attributes on the AD server.

```
# id DOMAIN\\administrator  
uid=10002(DOMAIN\\administrator) gid=10000(DOMAIN\\domain users)  
groups=10000(DOMAIN\\domain users)
```

Configuring Active Directory using Kerberos with RFC2307 ID mapping

The following steps provide an example of configuring Active Directory (AD) by using Kerberos with RFC2307 mapping.

1. Issue the **mmuserauth service create** command as shown in the following example:

```
# mmuserauth service create --data-access-method file --type ad --netbios-name  
kknnode_v42 --servers myADserver --user-name administrator --idmap-role master  
--enable-nfs-kerberos --unixmap-domains "DOMAIN(10000-20000)"
```

The system displays the following output:

```
File authentication configuration completed successfully.
```

2. Issue the **mmuserauth service list** to verify the authentication configuration as shown in the following example:

```
# mmuserauth service list
```

The system displays the following output:

```

FILE access configuration : AD
PARAMETERS          VALUES
-----
ENABLE_NFS_KERBEROS    true
SERVERS              "*"
USER_NAME             kknodel_v42$
NETBIOS_NAME          kknodel_v42
IDMAP_ROLE            master
IDMAP_RANGE           10000000-299999999
IDMAP_RANGE_SIZE      1000000
UNIXMAP_DOMAINS       DOMAIN(1000-20000:win)
LDAPMAP_DOMAINS      none

OBJECT access not configured
PARAMETERS          VALUES

```

- Verify the user name resolution on the system. Confirm that the resolution is showing IDs that are pulled from RFC2307 attributes on the AD server.

```
# id DOMAIN\administrator
uid=10002(DOMAIN\administrator) gid=40000(DOMAIN\domain users)
groups=11000545(BUILTIN\users),11000544 (BUILTIN\administrators)
```

Configuring Active Directory using IPv6 address with RFC2307 ID mapping

The following steps provide an example of configuring Active Directory (AD) by using IPv6 address with RFC2307 mapping.

- Issue the **mmuserauth service create** command as shown in the following example:

```
# mmuserauth service create --type ad --data-access-method file --servers [2001:192::e61f:122:feb7:5df0]
--netbios-name specscale --user-name adUser --idmap-role master --unixmap-domains
'TESTDOMAIN(10000-50000:win)'
```

The system displays the following output:

```
File authentication configuration completed successfully.
```

- Issue the **mmuserauth service list** to verify the authentication configuration as shown in the following example:

```
# mmuserauth service list
```

The system displays output similar to this:

```

FILE access configuration : AD
PARAMETERS          VALUES
-----
ENABLE_NFS_KERBEROS    false
SERVERS              "*"
USER_NAME             adUser$
NETBIOS_NAME          specscale
IDMAP_ROLE            master
IDMAP_RANGE           10000000-299999999
IDMAP_RANGE_SIZE      1000000
UNIXMAP_DOMAINS       TESTDOMAIN(10000-50000:win)
LDAPMAP_DOMAINS      none

OBJECT access not configured
PARAMETERS          VALUES
-----
```

AD authentication with RFC2307 ID mapping for picking UNIX primary group

You can configure IBM Storage Scale system authentication with Active Directory (AD) and RFC2307 ID mapping or AD with Kerberos NFS and RFC2307 ID mapping. In these authentication methods, use Active Directory to store user credentials and RFC2307 attributes on the same AD server to store UIDs and GIDs. These authentication schemes are useful when you are planning to use any pre-existing UNIX client or NFS protocol together with SMB protocols for data access. RFC2307 ID mapping is configurable per AD domain. If you use AD-based authentication and the ID maps are not configured with RFC2307, the IBM Storage Scale system uses the automatic ID mappings by default.

The following provides an example of how to configure the IBM Storage Scale system with Active Directory and RFC2307 ID mapping for picking UNIX primary group:

1. Submit the **mmuserauth service create** command as shown in the following example:

```
# mmuserauth service create --type ad --data-access-method file --netbios-name ess  
--user-name administrator --idmap-role master --servers myADserver --idmap-range-size  
1000000  
--idmap-range 10000000-299999999 --unixmap-domains 'DOMAIN(5000-20000:unix)'
```

The system displays this output:

```
File authentication configuration completed successfully.
```

2. Issue this command to verify the authentication configuration:

```
mmuserauth service list
```

The system displays the following output:

```
FILE access configuration : AD  
PARAMETERS          VALUES  
-----  
ENABLE_NFS_KERBEROS    false  
SERVERS                "*"  
USER_NAME              ess$  
NETBIOS_NAME           ess  
IDMAP_ROLE              master  
IDMAP_RANGE             10000000-299999999  
IDMAP_RANGE_SIZE        1000000  
UNIXMAP_DOMAINS         DOMAIN(5000-20000:unix)  
LDAPMAP_DOMAINS        none  
  
OBJECT access not configured  
PARAMETERS          VALUES
```

3. Verify the user name resolution on the system after successfully authenticating the user. Confirm that the resolution is showing primary group picked up as defined in UNIX attribute of the user. Validate the IDs that are pulled are from RFC2307 attributes on the AD server:

```
# id DOMAIN\unixuser
```

The system displays the following output:

```
uid=10002(DOMAIN\unixuser) gid=10000(DOMAIN\unix users)  
groups=10000(DOMAIN\unix users), 11000545(BUILTIN\users),11000544 (BUILTIN\administrators)
```

Configuring AD using Kerberos with RFC2307 ID mapping

1. Submit the **mmuserauth service create** command as shown in the following example:

```
# mmuserauth service create --data-access-method file --type ad --netbios-name ess --servers  
myADserver --user-name administrator --idmap-role master --enable-nfs-kerberos --unixmap-  
domains  
"DOMAIN(10000-200000:unix)"
```

The system displays the following output:

```
File authentication configuration completed successfully.
```

2. Issue the **mmuserauth service list** command to verify the authentication configuration as shown in the following example:

```
# mmuserauth service list
```

The system displays the following output:

```

FILE access configuration : AD
PARAMETERS          VALUES
-----
ENABLE_NFS_KERBEROS    true
SERVERS              "*"
USER_NAME             ess$
NETBIOS_NAME          ess
IDMAP_ROLE            master
IDMAP_RANGE           10000000-29999999
IDMAP_RANGE_SIZE      1000000
UNIXMAP_DOMAINS       DOMAIN(10000-20000:unix)
LDAPMAP_DOMAINS       none

OBJECT access not configured
PARAMETERS          VALUES

```

- Verify the user name resolution on the system after successfully authenticating the user. Confirm that the resolution is showing primary group picked up as defined in the UNIX attribute of the user. Validate the IDs that are pulled are from RFC2307 attributes on the AD server:

```
# id DOMAIN\unixuser
```

The system displays the following output:

```

uid=10002(DOMAIN\unixuser) gid=10000(DOMAIN\unix users)
groups=10000(DOMAIN\unix users), 11000545(BUILTIN\users),11000544 (BUILTIN\administrators)

```

Note: If the domain mapping is set to ":unix", it is expected that the user's unix attribute "gidNumber" is populated correctly. If the attribute is empty, authentication falls back to the user's primary group's "gidNumber".

AD authentication with RFC2307 ID mapping for overlapping unixmap domain ranges

You can configure IBM Storage Scale system authentication with Active Directory (AD) and RFC2307 ID mapping where ID ranges of multiple unixmap domains intersect.

In the RFC2307 ID mapping method, the user and group IDs are stored and managed in the AD server and these IDs are used by the IBM Storage Scale system during file access. The RFC2307 ID mapping method is used when you want to have multiprotocol access.

Note: Make sure that users and groups across all AD domains have unique UIDs and GIDs to avoid ID collisions.

The following steps provide an example of how to configure the IBM Storage Scale system with AD and RFC2307 ID mapping for overlapping ID ranges of unixmap domains:

- Issue the following command as shown in this example:

```

# mmuserauth service create --data-access-method file --type ad --servers myADserver --user-
name adUser
--netbios-name specscale --idmap-role master --unixmap-domains "DOMAIN1(2000-4000);"
DOMAIN2(2000-4000)"
--enable-overlapping-unixmap-ranges

```

The system displays this output:

```

Enter Active Directory User 'adUser' password:
Enabling Overlapping unixmap ranges. Make sure that UIDs and GIDs are unique in order to
avoid ACLs
or/and data access issues. See man mmuserauth for further details.

File authentication configuration completed successfully.

```

- Issue this command to verify the authentication configuration:

```
# mmuserauth service list
```

The system displays the following output:

```
# mmuserauth service list
FILE access configuration : AD
PARAMETERS          VALUES
-----
ENABLE_NFS_KERBEROS    false
SERVERS              "*"
USER_NAME             specscale$
NETBIOS_NAME          specscale
IDMAP_ROLE            master
IDMAP_RANGE           10000000-299999999
IDMAP_RANGE_SIZE      1000000
UNIXMAP_DOMAINS       DOMAIN1(2000-4000:win);DOMAIN2(2000-4000:win)
LDAPMAP_DOMAINS       none
```

- Verify the user name resolution on the system. Confirm that the resolution is showing IDs that are pulled from RFC2307 attributes on the AD server.

```
# id DOMAIN1\\administrator
uid=2001(DOMAIN1\\administrator) gid=2101(DOMAIN1\\domain users)
groups=2101(DOMAIN1\\domain users)
```

```
# id DOMAIN2\\administrator
uid=3001(DOMAIN2\\administrator) gid=3101(DOMAIN2\\domain users)
groups=3101(DOMAIN2\\domain users)
```

Configuring AD using Kerberos with RFC2307 ID mapping for overlapping unixmap ranges

- Issue the following command as shown in this example:

```
# mmuserauth service create --data-access-method file --type ad --servers myADserver --user-name adUser
--netbios-name specscale --idmap-role master --enable-nfs-kerberos --unixmap-domains "DOMAIN1(2000-4000); DOMAIN2(2000-4000)"
--enable-overlapping-unixmap-ranges
```

The system displays this output:

```
Enter Active Directory User 'adUser' password:
Enabling Overlapping unixmap ranges. Make sure that UIDs and GIDs are unique in order to
avoid ACLs
or/and data access issues. See man mmuserauth for further details.

File authentication configuration completed successfully.
```

- Issue this command to verify the authentication configuration:

```
# mmuserauth service list
```

The system displays the following output:

```
# mmuserauth service list
FILE access configuration : AD
PARAMETERS          VALUES
-----
ENABLE_NFS_KERBEROS    true
SERVERS              "*"
USER_NAME             specscale$
NETBIOS_NAME          specscale
IDMAP_ROLE            master
IDMAP_RANGE           10000000-299999999
IDMAP_RANGE_SIZE      1000000
UNIXMAP_DOMAINS       DOMAIN1(2000-4000:win);DOMAIN2(2000-4000:win)
LDAPMAP_DOMAINS       none
```

- Verify the user name resolution on the system. Confirm that the resolution is showing IDs that are pulled from RFC2307 attributes on the AD server.

```
# id DOMAIN1\\administrator
uid=2001(DOMAIN1\\administrator) gid=2101(DOMAIN1\\domain users)
groups=2101(DOMAIN1\\domain users)
```

```
# id DOMAIN2\\administrator
uid=3001(DOMAIN2\\administrator) gid=3101(DOMAIN2\\domain users)
groups=3101(DOMAIN2\\domain users)
```

Best practices for configuring AD with RFC2307 as the authentication method

It is recommended to adhere to the following best practices if you configure Active Directory (AD) with RFC2307 as the authentication method:

- Remove any internal ID mappings present in the system before you configure AD with RFC2307. Otherwise, the system might detect the internal ID mappings instead of the RFC2307 ID mapping and abort the operation with an error message. In such situations, you are expected to clean up the entire authentication and ID mapping by using the **mmuserauth service remove** and **mmuserauth service remove --idmapdelete** command and then reconfigure AD authentication and RFC2307 ID mapping.
- If data is already present on the system, a complete removal of the authentication and ID mapping can cause permanent loss of data access.
- Using UIDs and GIDs greater than 1000 can avoid an overlap of IDs used by end users, administrative users, and operating system component users of the IBM Storage Scale system.

You can use AD-based authentication and RFC2307 ID mapping if you want to use the AFM feature of the IBM Storage Scale system.

Limitations of the mmuserauth service create command while configuring AD with RFC2307

The **mmuserauth service create** command that is used to configure authentication has the following limitations:

- The **mmuserauth service create** command does not check the two-way trust between the host domain and the RFC2307 domain that is required for ID-mapping services to function properly. The customer is responsible for configuring the two-way trust relationship between these domains.
- The customer is responsible for installing RFC2307 on the desired Active Directory server, and for assigning UIDs to users and GIDs to groups. The command does not return an error if RFC2307 is not installed, or if a UID or GID is not assigned.

Configuring AD-based authentication with LDAP ID mapping

Active Directory (AD) authentication with Lightweight Directory Access Protocol (LDAP) ID mapping provides a way for IBM Storage Scale to read ID mappings from an LDAP server as defined in RFC 2307. The LDAP server must be a stand-alone LDAP server. Mappings must be provided in advance by the administrator by creating the user accounts in the AD server and the `posixAccount` and `posixGroup` objects in the LDAP server. The names in the AD server and in the LDAP server must be the same. This ID-mapping approach allows the continued use of existing LDAP authentication servers that store records in the RFC2307 format. The group memberships that are defined in the AD server are also accepted in the system.

In the following example, AD is configured with the LDAP ID mapping.

1. Submit the **mmuserauth service create** command as shown in the following example:

```
# mmuserauth service create --data-access-method file --type ad --servers myADserver
--user-name administrator --netbios-name specscale
--idmap-role master --ldmap-domains "DOMAIN1(type=stand-alone:range=1000-100000
:ldap_srv=myLDAPserver:usr_dn=ou=People,dc=example,dc=com:grp_dn=ou=Groups,dc=example,
dc=com:bind_dn=cn=manager,dc=example,dc=com:bind_dn_pwd=password)"
```

Note: The `bind_dn_pwd` cannot contain the following special characters: semicolon (;), colon (:), opening brace ('), or closing brace ')'.

A sample output is as follows:

```
File authentication configuration completed successfully.
```

2. Issue the **mmuserauth service list** to verify the authentication configuration as shown in the following example:

```
# mmuserauth service list
```

A sample output is as follows:

```
FILE access configuration : AD
PARAMETERS VALUES
-----
ENABLE_NFS_KERBEROS false
SERVERS "*"
USER_NAME specscale$
NETBIOS_NAME specscale
IDMAP_ROLE master
IDMAP_RANGE 10000000-299999999
IDMAP_RANGE_SIZE 1000000
UNIXMAP_DOMAINS none
LDAPMAP_DOMAINS DOMAIN1(type=stand-alone: range=1000-100000:
ldap_srv=myLDAPserver:usr_dn=ou=People,dc=example,dc=com:
grp_dn=ou=Groups,dc=example,dc=com:bind_dn=cn=manager,dc=example,dc=com)
```

3. Verify the user name resolution on the system. Confirm that the resolution is showing IDs that are pulled from LDAP attributes on the AD server.

```
# id DOMAIN\\administrator
uid=10002(DOMAIN\administrator) gid=10000(DOMAIN\domain users)
groups=10000(DOMAIN\domain users)
```

4. To configure an IBM Storage Scale system with Active Directory that has IPv6 address and LDAP ID mapping, issue the following command:

```
# mmuserauth service create --type ad --data-access-method file --servers [2001:192::e61f:122:feb7:5df0]
--netbios-name specscale
--user-name adUser --idmap-role master --ldapmap-domains "TESTDOMAIN(type=stand-alone:
range=1000-10000:ldap_srv=[2001:192::e61f:122:feb7:5bf0]:
usr_dn=ou=People,dc=example,dc=com:grp_dn=ou=Groups,dc=example,
dc=com:bind_dn=cn=ldapuser,dc=example,dc=com:bind_dn_pwd=password)"
```

A sample output is as follows:

```
File Authentication configuration completed successfully.
```

5. To verify the authentication configuration with Active Directory that has IPv6 address, issue the **mmuserauth service list** command as shown in the following example:

```
# mmuserauth service list
```

A sample output is as follows:

```
FILE access configuration : AD
PARAMETERS          VALUES
-----
ENABLE_NFS_KERBEROS    false
SERVERS              "*"
USER_NAME             adUser$
NETBIOS_NAME          specscale
IDMAP_ROLE            master
IDMAP_RANGE           10000000-299999999
IDMAP_RANGE_SIZE      1000000
UNIXMAP_DOMAINS       none
LDAPMAP_DOMAINS       TESTDOMAIN(type=stand-alone: range=1000-10000:
ldap_srv=[2001:192::e61f:122:feb7:5bf0]:usr_dn=ou=People,dc=example,dc=com:grp_dn=
ou=Groups,dc=example,dc=com:bind_dn=cn=ldapuser,dc=example,dc=com)

OBJECT access not configured
PARAMETERS          VALUES
-----
```

Configuring LDAP-based authentication for file access

Using LDAP-based authentication can be useful when you use an external LDAP server to store user information and user passwords. In this authentication method, you can use LDAP as the authentication

as well as the ID mapping server for both NFS and SMB. Appropriate SMB schema needs to be uploaded in the LDAP if you plan to have SMB access.

Based on the level of security, the following configurations are possible:

- LDAP with TLS
- LDAP with Kerberos
- LDAP with TLS and Kerberos
- LDAP

Using LDAP with TLS secures the communication between the IBM Storage Scale system and the LDAP server, assuming that the LDAP server is configured for TLS.

You can use LDAP with Kerberos for higher security reasons. Kerberos is a network authentication protocol that provides secured communication by ensuring passwords are not sent over the network to the system. LDAP with Kerberos is typically used where an MIT KDC infrastructure exists and you are using it for various Kerberized application or if you want to have NFS and SMB with Kerberized access for higher security reasons.

The LDAP server might need to handle the login requests and ID mapping requests from the client that uses SMB protocol. Usually, the ID mapping requests are cached and they do not contribute to the load on the LDAP server unless the ID mapping cache is cleared due to a maintenance action. If the LDAP server cannot handle the load or a high number of connections, then the response to the login requests is slow or it might time out. In such cases, users need to retry their login requests.

It is assumed that LDAP server is set up with the required schemas installed in it to handle the authentication and ID mapping requests. If you need to support SMB data access, LDAP schema must be extended to enable storing of additional attributes such as SID, Windows password hash to the POSIX user object.

Note: The IBM Storage Scale system must not be configured with any authentication method before using LDAP as the authentication system for file access.

See “[Integrating with LDAP server](#)” on page 295 for more information on the prerequisites for integrating LDAP server with the IBM Storage Scale system.

Configuring LDAP with TLS for file access

You can configure LDAP with TLS as the authentication method for file access. Using TLS with LDAP helps you to have a secure communication channel between the IBM Storage Scale system and LDAP server.

In the following example, LDAP is configured with TLS as the authentication method for file access.

1. Ensure that the CA certificate for LDAP server is placed under /var/mmfs/tmp directory with the name `ldap_cacert.pem`; specifically, on the protocol node where the command is run. Perform validation of CA cert availability with wanted name at required location as shown in the following example:

```
# stat /var/mmfs/tmp/ldap_cacert.pem
File: /var/mmfs/tmp/ldap_cacert.pem
Size: 2130 Blocks: 8 IO Block: 4096 regular file
Device: fd00h/64768d Inode: 103169903 Links: 1
Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)
Context: unconfined_u:object_r:user_tmp_t:s0
Access: 2015-01-23 12:37:34.088837381 +0530
Modify: 2015-01-23 12:16:24.438837381 +0530
Change: 2015-01-23 12:16:24.438837381 +0530
```

2. Issue the `mmuserauth service create` command as shown in the following example:

```
# mmuserauth service create --type ldap --data-access-method file
--servers myLDAPserver --base-dn dc=example,dc=com
--user-name cn=manager,dc=example,dc=com
--netbios-name ess --enable-server-tls
```

A sample output is as follows:

```
File authentication configuration completed successfully.
```

3. Issue the **mmuserauth service list** command to see the current authentication configuration as shown in the following example:

```
# mmuserauth service list
```

A sample output is as follows:

```
FILE access configuration : LDAP
PARAMETERS          VALUES
-----
ENABLE_SERVER_TLS    true
ENABLE_KERBEROS     false
USER_NAME            cn=manager,dc=example,dc=com
SERVERS              myLDAPserver
NETBIOS_NAME         ess
BASE_DN               dc=example,dc=com
USER_DN               none
GROUP_DN              none
NETGROUP_DN           none
USER_OBJECTCLASS      posixAccount
GROUP_OBJECTCLASS     posixGroup
USER_NAME_ATTRIB      cn
USER_ID_ATTRIB        uid
KERBEROS_SERVER       none
KERBEROS_REALM        none

OBJECT access not configured
PARAMETERS          VALUES
-----
```

4. Verify the user resolution on system present in LDAP:

```
# id ldapuser2
uid=1001(ldapuser2) gid=1001(ldapuser2) groups=1001(ldapuser2)
```

5. To configure an IBM Storage Scale system with LDAP that has TLS and IPv6 address, issue the following command:

```
# mmuserauth service create --type ldap --data-access-method file --servers [2001:192::e61f:122:feb7:5df0]
--base-dn dc=example,dc=com --user-name cn=ldapuser,dc=example,dc=com
--netbios-name specscale --enable-server-tls
```

A sample output is as follows:

```
File Authentication configuration completed successfully.
```

6. To verify the authentication configuration with LDAP that has TLS and IPv6 address, issue the **mmuserauth service list** command as shown in the following example:

```
# mmuserauth service list
```

A sample output is as follows:

```
FILE access configuration : LDAP
PARAMETERS          VALUES
-----
ENABLE_SERVER_TLS    true
ENABLE_KERBEROS     false
USER_NAME            cn=ldapuser,dc=example,dc=com
SERVERS              [2001:192::e61f:122:feb7:5df0]
NETBIOS_NAME         specscale
BASE_DN               dc=example,dc=com
USER_DN               none
GROUP_DN              none
NETGROUP_DN           none
USER_OBJECTCLASS      posixAccount
GROUP_OBJECTCLASS     posixGroup
USER_NAME_ATTRIB      cn
```

```

USER_ID_ATTRIB          uid
KERBEROS_SERVER         none
KERBEROS_REALM          none

OBJECT access not configured
PARAMETERS              VALUES
-----
```

Configuring LDAP with Kerberos for file access

You can configure LDAP with Kerberos as the authentication method for file access. Using Kerberos with LDAP provides more security for the communication channel between the IBM Storage Scale system and LDAP server.

Example for configuring LDAP with Kerberos as the authentication method for file access.

1. Ensure that the keytab file is also placed under the /var/mmfs/tmp directory with the name as krb5_scale.keytab on the node where the command is run. Perform validation of keytab file availability with a desired name at a required location:

```
# stat /var/mmfs/tmp/krb5_scale.keytab
  File: /var/mmfs/tmp/krb5_scale.keytab
  Size: 1490   Blocks: 8      IO Block: 4096   regular file
Device: fd00h/64768d Inode: 68252098      Links: 1
Access: (0600/-rw-----)  Uid: (     0/    root)  Gid: (     0/    root)
Context: unconfined_u:object_r:user_tmp_t:s0
Access: 2021-05-26 06:52:49.511820164 -0400
Modify: 2021-04-28 09:52:07.661820164 -0400
Change: 2021-05-26 05:15:09.837820164 -0400
Birth: -
```

2. Issue the **mmuserauth service create** command as shown in the following example:

```
# mmuserauth service create --type ldap --data-access-method file
--servers myLDAPserver --base-dn dc=example,dc=com
--user-name cn=manager,dc=example,dc=com
--netbios-name ess --enable-kerberos --kerberos-server myKerberosServer
--kerberos-realm example.com
```

A sample output is as follows:

```
File authentication configuration completed successfully.
```

3. Issue the **mmuserauth service list** command to see the current authentication configuration as shown in the following example:

```
# mmuserauth service list
```

A sample output is as follows:

```

FILE access configuration : LDAP
PARAMETERS          VALUES
-----
ENABLE_SERVER_TLS    false
ENABLE_KERBEROS     true
USER_NAME           cn=manager,dc=example,dc=com
SERVERS             myLDAPserver
NETBIOS_NAME        ess
BASE_DN              dc=example,dc=com
USER_DN              none
GROUP_DN             none
NETGROUP_DN          none
USER_OBJECTCLASS    posixAccount
GROUP_OBJECTCLASS   posixGroup
USER_NAME_ATTRIB    cn
USER_ID_ATTRIB       uid
KERBEROS_SERVER     myKerberosServer
KERBEROS_REALM      example.com

OBJECT access not configured
PARAMETERS          VALUES
-----
```

- To configure an IBM Storage Scale system with LDAP and Kerberos servers that have IPv6 address, issue the following command:

```
# mmuserauth service create --type ldap --data-access-method file --servers [2001:192::e61f:122:feb7:5df0] --base-dn dc=example,dc=com --user-name cn=ldapuser,dc=example,dc=com --netbios-name specscale --enable-kerberos --kerberos-server [2001:192::e61f:122:feb7:5dc0]
```

A sample output is as follows:

```
File Authentication configuration completed successfully.
```

- To verify the authentication configuration with LDAP and Kerberos servers that have IPv6 address, issue the **mmuserauth service list** command.

```
# mmuserauth service list
```

A sample output is as follows:

```
FILE access configuration : LDAP
PARAMETERS          VALUES
-----
ENABLE_SERVER_TLS    false
ENABLE_KERBEROS     true
USER_NAME           cn=ldapuser,dc=example,dc=com
SERVERS             [2001:192::e61f:122:feb7:5df0]
NETBIOS_NAME        specscale
BASE_DN              dc=example,dc=com
USER_DN              none
GROUP_DN             none
NETGROUP_DN          none
USER_OBJECTCLASS     posixAccount
GROUP_OBJECTCLASS   posixGroup
USER_NAME_ATTRIB    cn
USER_ID_ATTRIB      uid
KERBEROS_SERVER     [2001:192::e61f:122:feb7:5dc0]
KERBEROS_REALM      MYREALM.com

OBJECT access not configured
PARAMETERS          VALUES
-----
```

Configuring LDAP with TLS and Kerberos for file access

You can configure LDAP with TLS and Kerberos as an authentication method for file access. Using Kerberos and TLS with LDAP provides maximum security for the communication channel between the IBM Storage Scale system and the LDAP server.

Provides an example on how to configure LDAP with TLS and Kerberos as an authentication method for file access.

- Ensure that the CA certificate for LDAP server is placed under the /var/mmfs/tmp directory with the `ldap_cacert.pem` name. Specifically, on a protocol node where the command is run. Perform validation of CA cert availability with a desired name at a required location.

```
# stat /var/mmfs/tmp/ldap_cacert.pem
File: /var/mmfs/tmp/ldap_cacert.pem
Size: 2130 Blocks: 8 IO Block: 4096 regular file
Device: fd00h/64768d Inode: 103169903 Links: 1
Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)
Context: unconfined_u:object_r:user_tmp_t:s0
Access: 2015-01-23 12:37:34.088837381 +0530
Modify: 2015-01-23 12:16:24.438837381 +0530
Change: 2015-01-23 12:16:24.438837381 +0530
```

- Ensure that the keytab file is placed under /var/mmfs/tmp directory name as `krb5_scale.keytab` specifically on the node where the command is run. Perform validation of keytab file availability with a desired name at a required location:

```
# stat /var/mmfs/tmp/krb5_scale.keytab
File: /var/mmfs/tmp/krb5_scale.keytab
Size: 1490     Blocks: 8          IO Block: 4096   regular file
Device: fd00h/64768d Inode: 68252098    Links: 1
Access: (0600/-rw-----) Uid: ( 0/    root)  Gid: ( 0/    root)
```

```
Context: unconfined_u:object_r:user_tmp_t:s0
Access: 2021-05-26 06:52:49.511820164 -0400
Modify: 2021-04-28 09:52:07.661820164 -0400
Change: 2021-05-26 05:15:09.837820164 -0400
Birth: -
```

3. Issue the **mmuserauth service create** command.

```
# mmuserauth service create --type ldap --data-access-method file
--servers myLDAPserver --base-dn dc=example,dc=com
--user-name cn=manager,dc=example,dc=com
--netbios-name ess --enable-server-tls --enable-kerberos
--kerberos-server myKerberosServer --kerberos-realm example.com
```

A sample output is as follows:

```
File authentication configuration completed successfully.
```

4. To verify the authentication configuration, issue the **mmuserauth service list** command.

```
# mmuserauth service list
```

A sample output is as follows:

```
FILE access configuration : LDAP
PARAMETERS          VALUES
-----
ENABLE_SERVER_TLS    true
ENABLE_KERBEROS     true
USER_NAME            cn=manager,dc=example,dc=com
SERVERS              myLDAPserver
NETBIOS_NAME         ess
BASE_DN               dc=example,dc=com
USER_DN               none
GROUP_DN              none
NETGROUP_DN           none
USER_OBJECTCLASS      posixAccount
GROUP_OBJECTCLASS     posixGroup
USER_NAME_ATTRIB      cn
USER_ID_ATTRIB        uid
KERBEROS_SERVER       myKerberosServer
KERBEROS_REALM        example.com

OBJECT access not configured
PARAMETERS          VALUES
-----
```

5. Verify the user resolution on the system.

```
# id ldapuser3
uid=1002(ldapuser3) gid=1002(ldapuser3) groups=1002(ldapuser3)
```

Configuring LDAP without TLS and Kerberos for file access

You can configure LDAP without TLS or Kerberos for file access. But this method is less secured compared to LDAP with TLS, LDAP with TLS and Kerberos, and LDAP with Kerberos configurations.

The following information provides an example on how to configure LDAP without TLS and Kerberos as the authentication method for file access:

1. Issue the **mmuserauth service create** command.

```
# mmuserauth service create --type ldap --data-access-method file
--servers 192.0.2.18 --base-dn dc=example,dc=com
--user-name cn=manager,dc=example,dc=com --netbios-name ess
```

A sample output is as follows:

```
File Authentication configuration completed successfully.
```

2. To verify the authentication configuration, issue the **mmuserauth service list** command as shown in the following example:

```
# mmuserauth service list
```

A sample output is as follows:

```
FILE access configuration : LDAP
PARAMETERS          VALUES
-----
ENABLE_SERVER_TLS    false
ENABLE_KERBEROS     false
USER_NAME            cn=manager,dc=example,dc=com
SERVERS              192.0.2.18
NETBIOS_NAME         ess
BASE_DN               dc=example,dc=com
USER_DN               none
GROUP_DN              none
NETGROUP_DN           none
USER_OBJECTCLASS      posixAccount
GROUP_OBJECTCLASS     posixGroup
USER_NAME_ATTRIB      cn
USER_ID_ATTRIB        uid
KERBEROS_SERVER       none
KERBEROS_REALM        none

OBJECT access not configured
PARAMETERS          VALUES
-----
```

3. To configure an IBM Storage Scale system with LDAP that has IPv6 address, issue the following command:

```
# mmuserauth service create --type ldap --data-access-method file --servers [2001:192::e61f:122:feb7:5df0]
--base-dn dc=example,dc=com --user-name cn=ldapuser,dc=example,dc=com --netbios-name specscale
```

A sample output is as follows:

```
File Authentication configuration completed successfully.
```

4. To verify the authentication configuration with LDAP that has IPv6 address, issue the **mmuserauth service list** command.

```
# mmuserauth service list
```

A sample output is as follows:

```
FILE access configuration : LDAP
PARAMETERS          VALUES
-----
ENABLE_SERVER_TLS    false
ENABLE_KERBEROS     false
USER_NAME            cn=ldapuser,dc=example,dc=com
SERVERS              [2001:192::e61f:122:feb7:5df0]
NETBIOS_NAME         specscale
BASE_DN               dc=example,dc=com
USER_DN               none
GROUP_DN              none
NETGROUP_DN           none
USER_OBJECTCLASS      posixAccount
GROUP_OBJECTCLASS     posixGroup
USER_NAME_ATTRIB      cn
USER_ID_ATTRIB        uid
KERBEROS_SERVER       none
KERBEROS_REALM        none

OBJECT access not configured
PARAMETERS          VALUES
-----
```

Configuring NIS-based authentication

The Network Information Service (NIS)-based authentication is useful in NFS-only environment where NIS acts as an ID mapping server and also used for netgroups. When the file access is configured with NIS, SMB access cannot be enabled.

Ensure that you have the following details before you start NIS-based authentication:

- NIS domain name. This domain name is case-specific.
- IP address or host name of the NIS server
- Primary DNS is added in the `/etc/resolv.conf` file on all the protocol nodes. It resolves the authentication server system with which the IBM Storage Scale system is configured. The manual changes made in the configuration files might be overwritten by the operating system's network manager. Therefore, ensure that the DNS configuration is persistent even after you restart the system. For more information on the circumstances where the configuration files are overwritten, refer the corresponding operating system documentation.

NIS has many security weaknesses in contrast to current IT security standards. The default configuration of the NIS server is inherently insecure. The communication with the NIS server over RPC calls can be sniffed on the network. Because of these security risks, it is highly recommended to migrate to more secure directory server implementations such as LDAP or Active Directory. If the NIS infrastructure replacement is not feasible, refer the operating system documentation to secure the NIS server and the communication with the NIS server.

You need to run the **mmuserauth service create** command with the following mandatory parameters to configure NIS as the authentication method:

- `--type nis`
- `--data-access-method file`
- `--domain domainName`
- `--servers comma-delimited IP address or host name`

For more information on each parameter, see the **mmuserauth service create** command.

Note: NIS authentication is not supported for RHEL 9.

Provides an example on how to configure NIS as the authentication method for file access.

1. Issue the **mmuserauth service create** command as shown in the following example:

```
# mmuserauth service create --type nis --data-access-method file  
--servers myNISserver --domain nisdomain3
```

The system displays the following output:

```
File Authentication configuration completed successfully.
```

2. To verify the authentication configuration, issue the **mmuserauth service list** command as shown in the following example:

```
# mmuserauth service list
```

The system displays the following output:

```
FILE access configuration : NIS  
PARAMETERS          VALUES  
-----  
SOURCES             myNISserver  
DOMAIN              nisdomain3  
  
OBJECT access not configured  
PARAMETERS          VALUES  
-----.
```

Configuring file authentication by using GUI

You can configure an authentication method or view the existing authentication method that is used for Network File System (NFS) and Server Message Block (SMB) users from the **Services > File Authentication** page of the GUI.

The IBM Storage Scale system supports the following file user authentication methods to authenticate an NFS or SMB user:

Active Directory

Uses Microsoft Active Directory (AD) as the authentication server. This method is used if you need to authenticate SMB users to access the data through SMB shares. When you select AD as the authentication server, you need to configure an ID-mapping method to map the user IDs from the external domain with a set of internal user IDs. You can configure the following ID-mapping methods: Automatic ID mapping, RFC2307 ID mapping, and LDAP ID mapping. The details of these ID-mapping methods are explained in the procedure.

LDAP

Uses an LDAP server to authenticate users. This is the ideal method to authenticate the NFS protocol users to access the data through the NFS exports.

NIS

The NIS-based authentication is useful in NFS-only environment where NIS acts as an ID-mapping server and used for net groups. When file access is configured with NIS, SMB access cannot be enabled.

Note: NIS authentication is not supported for RHEL 9.

User-defined

The user can select the authentication and ID-mapping methods of their choice. It is the responsibility of the administrator of the client system to manage the authentication and ID mapping for file access to the IBM Storage Scale system.

Example for how to configure file authentication by using GUI

The following steps show how to configure an Active Directory-based file authentication method by using GUI:

1. Go to **Services > File Authentication** page in the IBM Storage Scale GUI.
2. Click **Configure File Authentication**. The Configure File Authentication wizard appears.
3. Select **Active Directory** as the authentication method from the following list of authentication methods:
 - Active Directory
 - Lightweight Directory Access Protocol (LDAP)
 - Network Information Service (NIS) for NFS
 - User-defined
4. Type the AD domain controller in the **Server** field.
5. Type the username in the **User name** field. This user name is used for initial access to the authentication server in the configuration phase.
6. Type the password for the user name in the **Password** field.
7. Select **Show password** if you want to verify the password that you entered.
8. Type the NetBIOS name in the **NetBIOS** name field. The NetBIOS name that is used for identifying the cluster in the AD. A machine account based on the NetBIOS name is created when this cluster joins AD. This account is used for communication between the cluster and AD.
9. Click **Next** and configure ID mapping. You can configure the following ID-mapping methods:
 - **Automatic ID mapping:** The user and group IDs are automatically generated and stored within the IBM Storage Scale system. When an external ID-mapping server is not present in the environment or cannot be used, then this ID-mapping method can be used. This method is typically used if you have SMB only access and do not plan to deploy multiprotocol access. That is, the AD-based

authentication with automatic ID mapping is not used when you need to allow NFS and SMB access to the same data.

- **RFC2307 ID mapping:** The user and group IDs are stored and managed in the AD server and these IDs are used by the IBM Storage Scale system during file access. The RFC2307 ID-mapping method is used when you want to have multiprotocol access. That is, you can have both NFS and SMB access over the same data.
- **LDAP ID mapping:** In the LDAP-mapping method, user ID and group ID are stored and managed in the LDAP server, and these IDs are used by the IBM Storage Scale system during file access. The LDAP ID-mapping method is used when you want to have multiprotocol access. That is, you can have both NFS and SMB access over the same data.

For Automatic ID mapping

10. Select the ID-mapping role from the **ID mapping role** field. You can select either **Master** or **Subordinate** as the ID map role. For **Master** role, the system creates the ID maps. If you select **Subordinate**, the system does not create ID maps on its own. In such cases, ID maps must be exported from the master to the subordinate. While using automatic ID mapping, to have same ID maps on systems that share AFM relationship, you need to export the ID mappings from master to subordinate.
11. In the **ID range** field, specify the range of values from which the IBM Storage Scale UIDs and GIDs are assigned by the system to the Active Directory users and groups. The default value is 10000000-299999999.
12. In the **ID map size** field, specify the range of values from which the IBM Storage Scale UIDs and GIDs are assigned by the system to the Active Directory users and groups. The lower value of the range must be at least 1000.
13. Click **Next** to configure RFC2307 ID mapping.

For RFC2307 ID mapping

14. Specify the AD domain for which the ID mapping needs to be configured, in the **Domain name** field.
15. In the **ID range** field, specify the range of users or groups from a domain that needs access to data exports.
16. Select the source of the primary group from the **Primary group source** field. You can select either Windows primary group of a user in the AD or the primary group as set in the UNIX attributes of a user in the AD.
17. Select the **Enable Kerberized logins** checkbox when you want to enable Kerberized login for the users who gain access by using NFSv3 or NFSv4 protocols.
18. Click **Next** to configure LDAP ID mapping.

For LDAP ID mapping

19. In the **Domain name** field, specify the AD domain for which ID-mapping service needs to be configured.
20. Specify the **LDAP server** that manages the ID mapping.
21. In the **ID range** field, specify the range of IDs from which the UID and GID must be assigned.
22. In the **User DN** field, specify the bind tree on the LDAP server where user objects are located.
23. In the **Group DN** field, specify the bind tree on the LDAP server where group objects are located.
24. In the **Bind DN** field, specify the user DN that must be used for authentication in the LDAP server. If not specified, anonymous bind is performed.
25. Specify the user DN password that is specified in the bind DN, in the **Bind password** field. Select the **Show password** checkbox if you want to verify the password that you entered.
26. Click **Next** to continue.
27. Review the details of the configuration in the **Summary** page of the **Configure File Authentication** wizard.
28. Select the **Test connection to the Active Directory server** checkbox, if you want to verify whether the AD server is reachable to all protocol nodes.

- Click **Finish** to complete the process. The system runs the commands in the background and completes the file authentication configuration and displays the status of the operation.

Viewing, modifying, or deleting the file authentication configuration

You can also perform the following tasks from the **Services > File Authentication** page in the GUI.

- View the existing configuration. The existing authentication is specified under the **Settings** tab.
- Modify the existing authentication configuration.
- Delete the existing configuration and ID mappings, if any.

Modifying or deleting an existing configuration can be done by using the **Edit** option that is available under the **Settings** tab of the **Services > File Authentication** page. This opens the **Configure File Authentication** wizard. Follow the wizard to either switch to a new authentication after clearing the existing configuration or only to remove the existing configuration.

Related concepts

Prerequisites

Ensure that the following requirements are met before you start configuring an authentication method for file access.

Configuring file authentication by using CLI

You need to use the **mmuserauth service create** command to configure user authentication by using CLI.

Configuring authentication for Swift Object access

Important:

- CES Swift Object protocol feature is not supported from IBM Storage Scale 5.2.0 onwards.
- IBM Storage Scale 5.1.8 is the last release that has CES Swift Object protocol.
- IBM Storage Scale 5.2.0 will tolerate the update of a CES node from IBM Storage Scale 5.1.8.
 - Tolerate* means:
 - The CES node will be updated to 5.2.0.
 - Swift Object support will not be updated as part of the 5.2.0 update.
 - You may continue to use the version of Swift Object protocol that was provided in IBM Storage Scale 5.1.8 on the CES 5.2.0 node.
 - IBM will provide usage and known defect support for the version of Swift Object that was provided in IBM Storage Scale 5.1.8 until you migrate to a supported object solution that IBM Storage Scale provides.
 - CES Swift Object is replaced with IBM Storage Scale S3. For more details, refer to the *S3 support overview* section the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
 - For more information about Swift Object in IBM Storage Scale, refer to the [IBM Storage Scale 5.2.0 documentation](#).
- Please contact IBM for further details and migration planning.

Related concepts

Setting up authentication servers to configure protocol user access

Before you start configuring authentication for protocol access, ensure that the authentication server is set up and the connection between the IBM Storage Scale system and authentication server is established.

Configuring authentication and ID mapping for file access

The system administrator can decide whether to configure authentication and ID mapping method either during the installation of the IBM Storage Scale system or after the installation. If the authentication configuration is not configured during installation, you can manually do it by using the **mmuserauth service create** command from any protocol node of the IBM Storage Scale system or by using the IBM Storage Scale management GUI.

Managing user-defined authentication

In the user-defined mode of authentication, the user is free to select the authentication and ID-mapping methods of their choice. It is the responsibility of the administrator of the client system to manage the authentication and ID mapping for file (NFS and SMB) and object access to the IBM Storage Scale system.

Listing the authentication configuration

Use the **mmuserauth service list** command to see the authentication method that is configured in the system.

Verifying the authentication services configured in the system

Use the **mmuserauth service check** command to check whether the authentication configuration is consistent across the cluster and the required services are enabled and running. This command validates and corrects the authentication configuration files and starts any associated services if needed.

Modifying the authentication method

If data already exists or is created with the existing authentication and ID-mapping method, it is not recommended to change the authentication or the ID-mapping modes. Changing the authentication method also might invalidate the existing ACLs that are applicable to files and directories. ACLs depend on the preexisting users and group IDs.

Deleting the authentication and the ID-mapping configuration

Deleting the authentication and ID-mapping configuration results in loss of access to data. Before you remove or edit ID mappings, determine how access to data is going to be maintained.

Authentication limitations

Consider the following authentication limitations when you configure and manage the IBM Storage Scale system:

Managing user-defined authentication

In the user-defined mode of authentication, the user is free to select the authentication and ID-mapping methods of their choice. It is the responsibility of the administrator of the client system to manage the authentication and ID mapping for file (NFS and SMB) and object access to the IBM Storage Scale system.

The IBM Storage Scale system administrators are not allowed use any of the GPFS commands to manage authentication. It is important for the end user to be aware of the limitations, if any, of the authentication and ID-mapping scheme that will be implemented after you configured the user-defined mode of authentication.

The user-defined mode is appropriate in the following circumstances:

- The client already has protocol deployments either on GPFS installations or on different systems and is planning to move to using the protocol stack on the IBM Storage Scale system. The client wants to replicate the current authentication and ID-mapping configuration. In this case, the client system administrator must be familiar with the required configuration settings that will be applied to the system.
- If the end user wants an authentication method that is not supported by the IBM Storage Scale system.

Note: If the end user wants to configure the authentication methods that are supported by the IBM Storage Scale system, it is highly recommended to configure the authentication and ID-mapping methods by using the **mmuserauth** command instead of opting for the user-defined method of authentication.

The IBM Storage Scale system administrator needs to specify that the user-defined mode of authentication is used by using the **--type userdefined** option in the **mmuserauth service create** command as shown in the following example:

```
# mmuserauth service create --type userdefined --data-access-method file  
File Authentication configuration completed successfully.
```

Submit the **mmuserauth service list** command to see the current authentication configuration as shown in the following example:

```
# mmuserauth service list  
FILE access configuration : USERDEFINED
```

PARAMETERS	VALUES
OBJECT access not configured	
PARAMETERS	VALUES

Typically, user-defined authentication is used when existing GPFS customers are already using GPFS with NFS and do not want to alter the authentication that is already configured on these systems. You can configure user-defined authentication for both object and file access or for object or file alone.

Note: Authorization depends upon authentication and ID mapping that is configured with the system. That is, the ACL control on exports, files, and directories depend on the authentication method that is configured.

File authentication configuration

Ensure the following while you are using the user-defined mode of authentication for file access:

- Ensure that the authentication server and ID-mapping server are always reachable from all the protocol nodes. For example, if NIS is configured as the ID-mapping server, you can use the 'ypwhich' command to ensure that NIS is configured and reachable from all the protocol nodes. Similarly, if LDAP is configured as authentication and ID-mapping server, you can bind to the LDAP server from all protocol nodes to monitor if the LDAP server is reachable from all protocol nodes.
- Ensure that the implemented authentication and ID-mapping configuration is always consistent across all the protocol nodes. This requires that the authentication server and ID-mapping server are manually maintained and monitored by the administrator. The administrator must also ensure that the configuration files are not overwritten due to node restart and other similar events.
- Ensure that the implemented authentication and ID mapping-related daemons and processes across the protocol nodes are always up and running.
- The users or groups, accessing the IBM Storage Scale system over NFS and SMB protocols must resolve to a unique UID and GID respectively on all protocol nodes. Especially, it must be resolved in implementations where different servers are used for authentication and ID mapping. The name that is registered in ID-mapping server for user and group must be checked for resolution.

For example,

```
# id fileuser
uid=1234(fileuser) gid=5678(filegroup) groups=5678(filegroup)
```

Note: However, in some use cases where only NFSV3 based access to the IBM Storage Scale system is used. In such cases, the user and group IDs are obtained from the NFS client and the ID-mapping setting is not configured on the protocol nodes.

- If the IBM Storage Scale system is configured for multiprotocol support (that is, the same data is accessed through both NFS and SMB protocols), ensure that the IDs of users and groups are consistent across the NFS clients and SMB clients and that they resolve uniquely on the protocol nodes.
- Ensure that UID and GID across users and groups that are accessing the system are not conflicting. This conflict check must be strictly enforced, especially in multiprotocol-based access deployments.
- Ensure that the Kerberos configuration files, placed on all protocol nodes, are in synchronization with each other. Ensure that the clients and the IBM Storage Scale system are part of the same Kerberos realm or trusted realm.
- While you are deploying two or more IBM Storage Scale clusters, ensure that the ID mapping is consistent in cases where you want to use IBM Storage Scale features like AFM, AFM-DR, and asynchronous replication of data.

The following table provides an overview of the authentication requirements for each file access protocol. Refer this table when you plan to use user-defined mode as the authentication method.

Table 22. Authentication requirements for each file access protocol.

File access protocol	Requirements
NFSv3	<p>In scenarios where user name and group name are expected to be used to native GPFS commands (for example, setting data ownership, listing user or group quota), the IBM Storage Scale system must be able to resolve the UID and GID to user name and group name and vice versa, consistently across all the protocol nodes.</p> <p>Note: However, for some use cases where only the NFSv3 based access to the IBM Storage Scale system is used. In such cases, the user and group IDs are coming from the NFS client and ID-mapping setting is not configured on the protocol nodes.</p>
Kerberos NFSv3	<p>Ensure that the user name and group name that are used to access data consistently resolve to same UID and GID across all protocol nodes and NFS clients.</p> <p>Ensure that the time is synchronized on the NFS server, NFS clients, and Kerberos server.</p> <p>Note: User names and group names are case-sensitive.</p>
NFSv4	<p>Ensure that the user name and group name that are used to access data consistently resolve to same UID and GID across all protocol nodes and NFS clients.</p> <p>Domain name must be specified in the /etc/idmapd.conf file and it must be the same on both the NFS server and NFS clients.</p> <p>Note: User names and group names are case-sensitive.</p>
Kerberos NFS V4	<p>Ensure that the user name and group name that are used to access data consistently resolve to same UID and GID across all protocol nodes and NFS clients.</p> <p>Ensure that the time is synchronized on the NFS server, NFS clients, and Kerberos server.</p> <p>Domain name and local-realms must be specified in the /etc/idmapd.conf file and it must be the same on both the NFS server and NFS clients.</p> <p>The value of "local-realms" takes the value of Kerberos realm with which the IBM Storage Scale system protocol nodes are configured.</p>

Table 22. Authentication requirements for each file access protocol. (continued)

File access protocol	Requirements
SMB	<p>Ensure that the user name and group name that are used to access data consistently resolve to same UID and GID across all protocol nodes and NFS clients.</p> <p>While you integrate with non-windows server, ensure that the samba attributes are populated on the directory server for every user and group that are planning to access the IBM Storage Scale system. Special care must be taken to match the samba domain SIDs.</p> <p>For Kerberized SMB access, ensure that time is synchronized the SMB server, SMB client, and Kerberos server.</p>

Related concepts

[Setting up authentication servers to configure protocol user access](#)

Before you start configuring authentication for protocol access, ensure that the authentication server is set up and the connection between the IBM Storage Scale system and authentication server is established.

[Configuring authentication and ID mapping for file access](#)

The system administrator can decide whether to configure authentication and ID mapping method either during the installation of the IBM Storage Scale system or after the installation. If the authentication configuration is not configured during installation, you can manually do it by using the **mmuserauth service create** command from any protocol node of the IBM Storage Scale system or by using the IBM Storage Scale management GUI.

[Configuring authentication for Swift Object access](#)

[Listing the authentication configuration](#)

Use the **mmuserauth service list** command to see the authentication method that is configured in the system.

[Verifying the authentication services configured in the system](#)

Use the **mmuserauth service check** command to check whether the authentication configuration is consistent across the cluster and the required services are enabled and running. This command validates and corrects the authentication configuration files and starts any associated services if needed.

[Modifying the authentication method](#)

If data already exists or is created with the existing authentication and ID-mapping method, it is not recommended to change the authentication or the ID-mapping modes. Changing the authentication method also might invalidate the existing ACLs that are applicable to files and directories. ACLs depend on the preexisting users and group IDs.

[Deleting the authentication and the ID-mapping configuration](#)

Deleting the authentication and ID-mapping configuration results in loss of access to data. Before you remove or edit ID mappings, determine how access to data is going to be maintained.

[Authentication limitations](#)

Consider the following authentication limitations when you configure and manage the IBM Storage Scale system:

Listing the authentication configuration

Use the **mmuserauth service list** command to see the authentication method that is configured in the system.

```
# mmuserauth service list
FILE access configuration : LDAP
PARAMETERS VALUES
-----
ENABLE_SERVER_TLS false
ENABLE_KERBEROS false
USER_NAME cn=manager,dc=example,dc=com
SERVERS 9.122.123.172
NETBIOS_NAME eslnode
BASE_DN dc=example,dc=com
USER_DN ou=people,dc=example,dc=com
GROUP_DN none
NETGROUP_DN ou=netgroup,dc=example,dc=com
USER_OBJECTCLASS inetOrgPerson
GROUP_OBJECTCLASS posixGroup
USER_NAME_ATTRIB cn
USER_ID_ATTRIB uid
KERBEROS_SERVER none
KERBEROS_REALM none
OBJECT access not configured
PARAMETERS VALUES
-----
```

For more information, see the topic *mmuserauth command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

[Setting up authentication servers to configure protocol user access](#)

Before you start configuring authentication for protocol access, ensure that the authentication server is set up and the connection between the IBM Storage Scale system and authentication server is established.

[Configuring authentication and ID mapping for file access](#)

The system administrator can decide whether to configure authentication and ID mapping method either during the installation of the IBM Storage Scale system or after the installation. If the authentication configuration is not configured during installation, you can manually do it by using the **mmuserauth service create** command from any protocol node of the IBM Storage Scale system or by using the IBM Storage Scale management GUI.

[Configuring authentication for Swift Object access](#)

[Managing user-defined authentication](#)

In the user-defined mode of authentication, the user is free to select the authentication and ID-mapping methods of their choice. It is the responsibility of the administrator of the client system to manage the authentication and ID mapping for file (NFS and SMB) and object access to the IBM Storage Scale system.

[Verifying the authentication services configured in the system](#)

Use the **mmuserauth service check** command to check whether the authentication configuration is consistent across the cluster and the required services are enabled and running. This command validates and corrects the authentication configuration files and starts any associated services if needed.

[Modifying the authentication method](#)

If data already exists or is created with the existing authentication and ID-mapping method, it is not recommended to change the authentication or the ID-mapping modes. Changing the authentication method also might invalidate the existing ACLs that are applicable to files and directories. ACLs depend on the preexisting users and group IDs.

[Deleting the authentication and the ID-mapping configuration](#)

Deleting the authentication and ID-mapping configuration results in loss of access to data. Before you remove or edit ID mappings, determine how access to data is going to be maintained.

Authentication limitations

Consider the following authentication limitations when you configure and manage the IBM Storage Scale system:

Verifying the authentication services configured in the system

Use the **mmuserauth service check** command to check whether the authentication configuration is consistent across the cluster and the required services are enabled and running. This command validates and corrects the authentication configuration files and starts any associated services if needed.

You can check the following authentication details by using the **mmuserauth service check** command:

- **--data-access-method {file | object | all}** Authentication method.
- **[-N|--nodes] {node-list | cesNodes}** Authentication configuration on each node. If the specified node is not a protocol node, the check operation is ignored on that node. If a protocol node is specified, then the system checks configuration on that protocol node. If you do not specify a node, the system checks the configuration of only the current node. To check authentication configuration on all protocol nodes, specify -N cesnodes.
- **--server-reachability** Verify whether the authentication backend server is reachable. If Swift Object is configured with external Keystone server, this check is not performed.
- **[-r | --rectify]** Rectify the configuration for the specified nodes by copying any missing configuration files or SSL/TLS certificates from another node.

For more information, see the *mmuserauth command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Example – File authentication check

Issue the **mmuserauth service check** command.

```
# mmuserauth service check --data-access-method file --nodes dgnode3,dgnode2 --server-reachability -r
dgnode2: not CES node. Ignoring...

Userauth file check on node: dgnode3
Checking SSSD_CONF: OK
Checking nsswitch file: OK
Checking Pre-requisite Packages: OK

LDAP servers status
LDAP server 192.168.122.250 : OK
Service 'sssd' status: OK
```

You can use the **id** command to see the list of users and groups that are fetched from the LDAP server. For example,

```
# id ldapuser2
uid=1001(ldapuser2) gid=1001(ldapuser2) groups=1001(ldapuser2)
```

Example - Swift Object authentication check

Issue the **mmuserauth service check** command.

```
# mmuserauth service check --server-reachability --data-access-method object
Userauth object check on node: dgnode3
Checking keystone.conf: OK
LDAP servers status
LDAP server sonash1 : OK
Service 'keystone-all' status: OK
```

Related concepts

[Setting up authentication servers to configure protocol user access](#)

Before you start configuring authentication for protocol access, ensure that the authentication server is set up and the connection between the IBM Storage Scale system and authentication server is established.

Configuring authentication and ID mapping for file access

The system administrator can decide whether to configure authentication and ID mapping method either during the installation of the IBM Storage Scale system or after the installation. If the authentication configuration is not configured during installation, you can manually do it by using the **mmuserauth service create** command from any protocol node of the IBM Storage Scale system or by using the IBM Storage Scale management GUI.

Configuring authentication for Swift Object access

Managing user-defined authentication

In the user-defined mode of authentication, the user is free to select the authentication and ID-mapping methods of their choice. It is the responsibility of the administrator of the client system to manage the authentication and ID mapping for file (NFS and SMB) and object access to the IBM Storage Scale system.

Listing the authentication configuration

Use the **mmuserauth service list** command to see the authentication method that is configured in the system.

Modifying the authentication method

If data already exists or is created with the existing authentication and ID-mapping method, it is not recommended to change the authentication or the ID-mapping modes. Changing the authentication method also might invalidate the existing ACLs that are applicable to files and directories. ACLs depend on the preexisting users and group IDs.

Deleting the authentication and the ID-mapping configuration

Deleting the authentication and ID-mapping configuration results in loss of access to data. Before you remove or edit ID mappings, determine how access to data is going to be maintained.

Authentication limitations

Consider the following authentication limitations when you configure and manage the IBM Storage Scale system:

Modifying the authentication method

If data already exists or is created with the existing authentication and ID-mapping method, it is not recommended to change the authentication or the ID-mapping modes. Changing the authentication method also might invalidate the existing ACLs that are applicable to files and directories. ACLs depend on the preexisting users and group IDs.

To modify the authentication method, do the following steps:

1. List the existing authentication configuration for file and Swift Object authentication method by using the **mmuserauth service list** command.
2. Identify the parameters that you need to change. If an authentication method and ID maps are existing, you must not plan to change the authentication type or ID-mapping schemes. When you remove the existing authentication method and ID maps, the user and group of users who were accessing the data cannot access the data anymore.

The following list provides the parameters that can be modified in each authentication configuration.

For file authentication:

- With LDAP authentication, all attributes of the configuration can be modified. When you change authentication servers, ensure that the newly specified servers are the replica of the original servers, otherwise, it might result in loss of access to data.
- With AD authentication, all attributes of the configuration can be modified. When you change the authentication server, ensure that the newly specified server is a domain controller in the same AD domain that is being served by the original server. Otherwise, it might result in loss of access to data. If UNIX ID maps are specified in current configuration and more new AD domains are to be added, it is vital to specify the current list of domains along with the new domains.

- With NIS authentication, all attributes of the configuration can be modified. When you change servers, ensure that the newly specified servers are serving the same NIS domain as the original servers; otherwise, it might result in loss of access to data.

For Swift Object authentication:

You can change all options except **--data-access-method** and **--type** parameters.

- Clean up the existing authentication by using the **mmuserauth service remove** command. Do not specify the **--idmapdelete** option as it results in loss of access to data.
- Issue the **mmuserauth service create** with the required parameter change; ensuring that you use the same authentication, ID-mapping scheme, and associated authentication servers.
- List the authentication configuration by using the **mmuserauth service list** to verify the change.
- Ensure that the authentication is consistent across the cluster by using the **mmuserauth service check** command.

Related concepts

[Setting up authentication servers to configure protocol user access](#)

Before you start configuring authentication for protocol access, ensure that the authentication server is set up and the connection between the IBM Storage Scale system and authentication server is established.

[Configuring authentication and ID mapping for file access](#)

The system administrator can decide whether to configure authentication and ID mapping method either during the installation of the IBM Storage Scale system or after the installation. If the authentication configuration is not configured during installation, you can manually do it by using the **mmuserauth service create** command from any protocol node of the IBM Storage Scale system or by using the IBM Storage Scale management GUI.

[Configuring authentication for Swift Object access](#)

[Managing user-defined authentication](#)

In the user-defined mode of authentication, the user is free to select the authentication and ID-mapping methods of their choice. It is the responsibility of the administrator of the client system to manage the authentication and ID mapping for file (NFS and SMB) and object access to the IBM Storage Scale system.

[Listing the authentication configuration](#)

Use the **mmuserauth service list** command to see the authentication method that is configured in the system.

[Verifying the authentication services configured in the system](#)

Use the **mmuserauth service check** command to check whether the authentication configuration is consistent across the cluster and the required services are enabled and running. This command validates and corrects the authentication configuration files and starts any associated services if needed.

[Deleting the authentication and the ID-mapping configuration](#)

Deleting the authentication and ID-mapping configuration results in loss of access to data. Before you remove or edit ID mappings, determine how access to data is going to be maintained.

[Authentication limitations](#)

Consider the following authentication limitations when you configure and manage the IBM Storage Scale system:

Deleting the authentication and the ID-mapping configuration

Deleting the authentication and ID-mapping configuration results in loss of access to data. Before you remove or edit ID mappings, determine how access to data is going to be maintained.

Removing file authentication

Note: You are not allowed to delete both the authentication configuration and the ID mappings at the same time. You need to remove the authentication configuration first and then the ID maps. Because of this system, you cannot delete the ID maps without deleting the authentication configuration.

1. Issue the **mmuserauth service list** command to see the authentication method that is configured in the system:

```
# mmuserauth service list
FILE access configuration: LDAP
PARAMETERS VALUES
-----
ENABLE_ANONYMOUS_BIND false
ENABLE_SERVER_TLS false
ENABLE_KERBEROS false
USER_NAME cn=manager,dc=example,dc=com
SERVERS 10.0.100.121
NETBIOS_NAME eslnode
BASE_DN dc=example,dc=com
USER_DN ou=people,dc=example,dc=com
GROUP_DN none
NETGROUP_DN ou=netgroup,dc=example,dc=com
USER_OBJECTCLASS inetOrgPerson
GROUP_OBJECTCLASS posixGroup
USER_NAME_ATTRIB cn
USER_ID_ATTRIB uid
KERBEROS_SERVER none
KERBEROS_REALM none
OBJECT access not configured
PARAMETERS VALUES
-----
```

2. Issue the **mmuserauth service remove** command to remove the authentication configuration as shown in the following example:

```
# mmuserauth service remove --data-access-method file
mmcesuserauth service remove: Command successfully completed.
```

3. Issue the **mmuserauth service list** command to verify whether the authentication configuration is removed:

```
# mmuserauth service list
FILE access not configured
PARAMETERS VALUES
-----
OBJECT access not configured
PARAMETERS VALUES
-----
```

For more information, see **mmuserauth** command in the *IBM Storage Scale: Command and Programming Reference Guide*.

Deleting authentication configuration as shown in the previous example does not delete the ID maps. Use the **--idmapdelete** option with the **mmuserauth service remove** command to remove ID maps that are created for user authentication:

```
# mmuserauth service remove --data-access-method file --idmapdelete
mmuserauth service remove: Command successfully completed
```

Related concepts

[Setting up authentication servers to configure protocol user access](#)

Before you start configuring authentication for protocol access, ensure that the authentication server is set up and the connection between the IBM Storage Scale system and authentication server is established.

[Configuring authentication and ID mapping for file access](#)

The system administrator can decide whether to configure authentication and ID mapping method either during the installation of the IBM Storage Scale system or after the installation. If the authentication configuration is not configured during installation, you can manually do it by using the **mmuserauth service create** command from any protocol node of the IBM Storage Scale system or by using the IBM Storage Scale management GUI.

[Configuring authentication for Swift Object access](#)

[Managing user-defined authentication](#)

In the user-defined mode of authentication, the user is free to select the authentication and ID-mapping methods of their choice. It is the responsibility of the administrator of the client system to manage the authentication and ID mapping for file (NFS and SMB) and object access to the IBM Storage Scale system.

[Listing the authentication configuration](#)

Use the **mmuserauth service list** command to see the authentication method that is configured in the system.

[Verifying the authentication services configured in the system](#)

Use the **mmuserauth service check** command to check whether the authentication configuration is consistent across the cluster and the required services are enabled and running. This command validates and corrects the authentication configuration files and starts any associated services if needed.

[Modifying the authentication method](#)

If data already exists or is created with the existing authentication and ID-mapping method, it is not recommended to change the authentication or the ID-mapping modes. Changing the authentication method also might invalidate the existing ACLs that are applicable to files and directories. ACLs depend on the preexisting users and group IDs.

[Authentication limitations](#)

Consider the following authentication limitations when you configure and manage the IBM Storage Scale system:

[**Authentication limitations**](#)

Consider the following authentication limitations when you configure and manage the IBM Storage Scale system:

File access limitations

AD based authentication

NFS with server-side group lookup and Active Directory authentication is only supported for Kerberized NFS access. The reason behind this is that obtaining the group membership of a user on a CES node is only possible after you authenticate the user authenticated on that node. With SMB, each new session is authenticated initially, which is sufficient to provide that information. With NFS, only Kerberized access can reliably provide the required information when you are using the Active Directory.

The following limitations exist for AD with automatic ID mapping:

- No support is provided for migrating the internally generated user and group ID maps to an external ID-mapping server. If data is stored on the IBM Storage Scale system with AD and automatic ID mapping, adding RFC2307 later requires the UIDs and GIDs. These UIDs and GIDs are used internally by the IBM Storage Scale system match the UIDs and GIDs that are stored in RFC2307. Matching is not possible if UIDs and GIDs, which are conflicting, are already stored in RFC2307. To avoid potential conflicts, configure the IBM Storage Scale system by using AD and RFC2307 from the beginning.
- Although AD along with automatic ID mapping can be used to have the same ID maps between systems that are in AFM relationship, this configuration is not a complete replacement for RFC2307. This configuration can be used in a predominantly SMB only setup, where NFS users are not already present in the environment. If NFS users are preexisting in the customer environment and these users intend to access the data with SMB users, then RFC2307 is mandatory.
- When AD-based authentication is used, SMB protocol access is kerberized by default. Access the system by using the netbios name that is specified in the command.

The following limitations exist for AD with RFC2307:

- Enabling RFC2307 for a trusted domain requires a two-way trust between the native and the trusted domains.
- To access the IBM Storage Scale system, users and groups must have a valid UID/GID assigned to them in AD. For user access, the windows group membership is evaluated on the IBM Storage Scale system. Hence, accessing a user's primary group is considered as the Microsoft Windows Primary group and not

the UNIX primary group that is listed in the UNIX attribute tab in the user's properties. Therefore, the user's primary Microsoft Windows group must be assigned with a valid GID.

- The **mmuserauth service create** command does not check the two-way trust between the native domain and the RFC2307 domain that is required for ID-mapping services to function properly. The customer is responsible for configuring the two-way trust relationship between these domains. The customer is responsible for assigning UIDs to users and GIDs to groups. The command does not return an error if a UID or GID is not assigned.

System Security Services Daemon (SSSD) should not be running on CES nodes when AD-based authentication is used. The AD-based authentication uses samba winbind process. Enabling SSSD when winbind is running on CES nodes might create library conflict. This conflict might affect the SSSD and/or IBM Storage Scale authentication.

LDAP-based authentication

The following limitations exist for LDAP-based authentication:

- Users with the same username from different organizational units under the specified baseDN in the LDAP server are denied access to SMB shares irrespective of the LDAP user suffix and LDAP group suffix values configured on the system.
- If multiple LDAP servers are specified during configuration, at any point in time, only one LDAP server is used.
- LDAP referrals are not supported.
- ACL management through windows clients is not supported.
- Only LDAP servers that implement RFC2307 schema are supported. IBM Storage Scale Protocol LDAP authentication is verified and tested against Linux OpenLDAP server. Other LDAP servers such as FreeIPA, RedHat IDM might work but are not Certified with IBM Storage Scale.

NIS-based authentication

- NIS configuration with an IPv6 address is not supported.
- NIS authentication is not supported for RHEL 9.

General limitations for file access

The following general limitations exist:

- When the SMB service is stopped on a protocol node, with any AD-based authentication method, the NFS-based access is also affected on that protocol node.
- When Microsoft Active Directory (AD) is used as an authentication system, the IBM Storage Scale system supports only the NetBIOS logon name for authentication and not the User Principle Name (UPN). Active Directory replaces some of the special characters that are used in the UPN with the underscore character (hexadecimal value 0x5F) for the related NetBIOS logon name of the user. For the complete list of the special characters that are replaced in the NetBIOS logon name, see Microsoft Active Directory documentation. Follow these steps to locate the NetBIOS logon name for an Active Directory domain user:

1. From the Windows Start menu, select **Administrative Tools > Active Directory Users and Computers**.
 2. Right-click the **Active Directory Domain user** for which you require the **NetBIOS** logon name.
 3. Select **Properties > Account** tab and check the value of the **User logon** name field (pre-Windows 2000).
- Authentication configuration commands restart the IBM Storage Scale protocol services such as SMB and NFS. The protocol services resume a few seconds after an authentication configuration command completes.
 - For file data access, switching or migrating from one authentication method to another is not supported because it might lead to loss of access to the data on the system.

- The IBM Storage Scale system does not support authentication servers (AD, LDAP, and NIS) that are running on virtual machines that are stored on an SMB or NFS export. The IBM Storage Scale system requires the authentication server to be running while you are configuring authentication and while the server is handling connection requests over protocols. The virtualizer cannot boot the authentication server unless the protocols are configured for authentication and data is ready to be served over the exports.
- The length of a username or a group name of the users and group of users who need to access the data cannot be more than 32 characters.
- The NFSV4 clients must be configured with the same authentication and ID-mapping server as the IBM Storage Scale system. The IBM Storage Scale system does not support an NFSV4 client that is configured with different authentication and ID-mapping servers.
- AIX clients follow a different methodology to integrate with AD. Therefore, NFSV4-based access from AIX clients to IBM Storage Scale is not supported when CES services are configured for AD and variations of AD-based authentication schemes.
- Based on the hardware platform that the protocol nodes are configured on, consider the group ID resolution in relation to the limitation that is described in the IBM Storage Scale FAQ. For more information, see [IBM Storage Scale FAQs](#).
- Regarding to the AD-based authentication scheme, the following considerations apply to configuring an NFS server to look up group membership information for an accessing NFS user:
 - The server-side group lookup function, which is enabled by setting the `MANAGE_GIDS` flag in the NFS configuration, works only after the user makes a valid authentication connection over CIFS.
 - You must make a valid authentication connection to the protocol node that serves the public IP from which the NFS export is to be mounted.
 - If the group membership of the user on an AD server changes, you must make a new valid CIFS connection to the protocol node. The protocol node serves the public IP from which the NFS export is to be mounted. This new connection reflects the changes on the protocol node of the CES cluster.
 - It is a good practice to make a valid authentication connection over CIFS to all the protocol nodes that participate in group membership evaluations. This practice results in uniform membership evaluations on all the protocol nodes of the CES cluster.
- To use NFSV4 ID mapping, you must set the NFS ID map domain on the IBM Storage Scale protocol nodes and you must configure the same NFS ID map domain on every NFS client. The following example demonstrates how to configure NFSV4 ID mapping.

- Issue the **`mnnfs config list`** command.

The system displays the following output, which shows that the ID map domain is not set:

```
Idmapd Configuration
=====
=====
```

- Enter the following command to set the NFS ID map domain:

```
# mnnfs config change IDMAPPED_DOMAIN=MY_IDMAP_DOMAIN
```

- Issue the **`mnnfs config list`** command to verify that the ID map domain is set.

The system displays this output:

```
Idmapd Configuration
=====
DOMAIN: MY_IDMAP_DOMAIN
=====
```

Related concepts

[Setting up authentication servers to configure protocol user access](#)

Before you start configuring authentication for protocol access, ensure that the authentication server is set up and the connection between the IBM Storage Scale system and authentication server is established.

Configuring authentication and ID mapping for file access

The system administrator can decide whether to configure authentication and ID mapping method either during the installation of the IBM Storage Scale system or after the installation. If the authentication configuration is not configured during installation, you can manually do it by using the **mmuserauth service create** command from any protocol node of the IBM Storage Scale system or by using the IBM Storage Scale management GUI.

Configuring authentication for Swift Object access

Managing user-defined authentication

In the user-defined mode of authentication, the user is free to select the authentication and ID-mapping methods of their choice. It is the responsibility of the administrator of the client system to manage the authentication and ID mapping for file (NFS and SMB) and object access to the IBM Storage Scale system.

Listing the authentication configuration

Use the **mmuserauth service list** command to see the authentication method that is configured in the system.

Verifying the authentication services configured in the system

Use the **mmuserauth service check** command to check whether the authentication configuration is consistent across the cluster and the required services are enabled and running. This command validates and corrects the authentication configuration files and starts any associated services if needed.

Modifying the authentication method

If data already exists or is created with the existing authentication and ID-mapping method, it is not recommended to change the authentication or the ID-mapping modes. Changing the authentication method also might invalidate the existing ACLs that are applicable to files and directories. ACLs depend on the preexisting users and group IDs.

Deleting the authentication and the ID-mapping configuration

Deleting the authentication and ID-mapping configuration results in loss of access to data. Before you remove or edit ID mappings, determine how access to data is going to be maintained.

Chapter 31. Managing protocol data exports

You can manage the data exports that you created by using NFS, SMB, and Object.

Managing SMB shares

SMB administration commands can be run from any cluster node, including non-CES nodes. However, the latency of the administration command execution on a CES node is lower because administrative changes are made immediately. Use the following information to manage SMB shares in IBM Storage Scale.

GUI navigation

To work with this function in the GUI, log on to the IBM Storage Scale GUI and select **Protocols > SMB Shares**.

Creating SMB share

Use the following information to create an SMB share:

1. Create the directory to be exported through SMB:

Note: IBM recommends an independent fileset for SMB shares.

Create a new independent fileset with these commands:

```
mmcrfileset fs01 fileset --inode-space=new  
mmlinkfileset fs01 fileset -J /gpfs/fs01/fileset
```

If the directory to be exported does not exist, create the directory first by running the following command:

```
mkdir /gpfs/fs01/fileset/smb
```

2. The recommended approach for managing access to the SMB share is to manage the ACLs from a Windows client machine. To change the ACLs from a Windows client, change the owner of the share folder to a user ID that will be used to make the ACL changes by running the following command:

```
chown 'DOMAIN\smbadmin' /gpfs/fs01/fileset/smb
```

3. Create the actual SMB share on the existing directory:

```
mmsmb export add smbexport /gpfs/fs01/fileset/smb
```

Additional options can be set during share creation. For a list of all the supported SMB options, see *mmsmb command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

4. Verify that the share has been created:

```
mmsmb export list
```

5. Access the share from a Windows client using the user ID that has been previously made the owner of the folder.
6. Right-click the folder in the Windows Explorer, open the **Security** tab, click **Advanced**, and modify the Access Control List as required.

Note: An SMB share can only be created when the ACL setting of the underlying file system is **-k nfs4**. In all other cases, **mmsmb export add** will fail with an error.

See “[Authorizing protocol users](#)” on page 474 for details and limitations.

GUI navigation

To work with this function in the GUI, log on to the IBM Storage Scale GUI and select **Protocols > SMB Shares**.

Creating an SMB share by using GUI

You can create an SMB share by using the IBM Storage Scale GUI to share data with the users of the system.

You need to enable and start the SMB service on the cluster to create an SMB share in the cluster. You can start and stop the SMB service from the **Services > SMB** page of the IBM Storage Scale GUI.

Perform the following steps to create an SMB share to host or share the data over the SMB protocol:

1. Go to **Protocols > SMB Shares** page in the IBM Storage Scale GUI. The SMB Shares page appears.
2. Click **Create Share**. The Create Share window appears. You can create a share either in the *Basic* mode or *Custom* mode. This procedure describes the steps to create an SMB share in the Custom mode.
3. Click **Custom** tab on the Create Share window.
4. Click **Browse** and select the path of the share in the **Path** field.
5. Type the name of the share in the **Share name** field.
6. Type the owner of the share in the **Owner** field.
7. Specify the associated owner group for the path in the **Owning group** field.
8. Click **Edit** to modify the file system ACL of the path that is going to be shared by the SMB protocol. This action only modifies the file system permission. SMB Share ACLs can be set only by using the **mmsmb** command.
9. Type a meaningful description in the **Comment** field.
10. Specify the SMB protocol-specific attributes to define share access, cross-protocol integration, oplocks, offline availability, encryption, data integrity, and permission to allow recalls from external pools in the corresponding fields. Hover help is available for each of these fields to explain about the features associated with these fields.
11. Specify the administrator user name in the **Administrative user** field. The administrative users have effective root level access through SMB and these users are not bound to the ACL permissions. You can specify multiple users as administrators by using comma after typing each user name.
12. Click **Create** to complete the SMB share creation.

Changing SMB share configuration

Use the following information to change the SMB share configurations.

For the documentation of all supported options, see *mmsmb command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

To see a list of supported configuration options for SMB shares, run the command:

```
mmsmb export list --key-info supported
```

For example, to change the descriptive comment for a share, run the command:

```
mmsmb export change smbshare --option 'comment=Project X export'
```

To list the configuration of all SMB shares, run the command:

```
mmsmb export list --all
```

Note: Changes to SMB share configurations only apply to client connections that have been established after the change has been made.

GUI navigation

To work with this function in the GUI, log on to the IBM Storage Scale GUI and select **Protocols > SMB Shares**.

Creating SMB share ACLs

The SMB protocol supports a separate level of ACLs that can be optionally added to an SMB share.

For more information, see [Managing ACLs of SMB exports using MMC](#).

SMB share ACLs can be added on the command line, as follows:

```
# mmsmb exportacl  
mmsmb exportacl: Missing arguments.  
Usage:  
mmsmb exportacl getid Retrieve the ID of user, group or system for use with SMB export ACLs.  
mmsmb exportacl list List SMB export ACLs.  
mmsmb exportacl add Add SMB export ACLs.  
mmsmb exportacl change Change SMB export ACLs.  
mmsmb exportacl remove Remove SMB export ACLs.  
mmsmb exportacl replace Replace SMB export ACLs.  
mmsmb exportacl delete Delete SMB export ACLs.
```

Examples:

1. %> mmsmb exportacl list smbexport

[smbexport]
ACL:\Everyone:ALLOWED/FULL
ACL:MYDOM06\Administrator:ALLOWED/FULL

2. %> mmsmb exportacl remove smbexport --user "\Everyone"

[smbexport]
ACL:MYDOM06\Administrator:ALLOWED/FULL

For details, see the information about managing the SMB share ACLs from a Windows client through the MMC.

Removing SMB shares

To remove an SMB share, use the **mmsmb** command. Use the following information to remove SMB shares:

1. Run the following command:

```
mmsmb export remove smbexport
```

2. Verify that the export has been removed by listing the configured SMB share again:

```
mmsmb export list
```

GUI navigation

To work with this function in the GUI, log on to the IBM Storage Scale GUI and select **Protocols > SMB Shares**.

Listing SMB shares

To list the SMB shares, run the following command:

```
mmsmb export list
```

Managing SMB shares using MMC

Microsoft Management Console (MMC) is a Windows tool that can be used to do basic configuration tasks on an SMB server. These tasks include administrative tasks such as listing or closing the connected users and open files, and creating and manipulating SMB shares. You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for managing SMB shares on the IBM Storage Scale cluster.



Attention: Listing a large number of entities (thousands of files, connections, locks, etc.) using Microsoft Management Console (MMC) might take a very long time and it might impact the performance of the file server. In these cases, it is recommended to use server-side administration tools. In certain cases like listing a very large number of open files, the MMC might also time-out and show no results if the server takes too long to collect the corresponding information.

Ensure that the following tasks are complete before you manage SMB shares:

- IBM Storage Scale is installed and configured.
- The SMB protocol is enabled and healthy SMB services are running on all protocol nodes.
- Required SMB shares are created and mounted from the Windows client.
- Microsoft Active Directory (AD) based authentication is set up. This includes:
 - Cluster nodes and client are domain members.
 - The client on which Microsoft Management Console (MMC) is running is a domain member.
 - Accurate DNS information is configured. If active sessions are listed, MMC tries to do a reverse pointer record lookup with DNS for every session (client IP), and if that fails then MMC hangs.
 - Involved NetBIOS names can be resolved using DNS.

For using the Shared Folders Microsoft Management Console (MMC) snap-in, you must be a member of the local administrators group of the cluster. After joining the cluster to an AD domain, only the domain admins group is a member of the administrators group of the cluster.

To add other users who can use the Shared Folders Microsoft Management Console (MMC) snap-in:

1. Connect to MMC as a user that is a member of the domain admins group.
2. Navigate to **System Tools > Local Users and Groups** and add a user to the local administrators group.

For more information, see the Microsoft Management Console documentation.

The following MMC features are not supported for managing SMB shares on the IBM Storage Scale cluster:

- Audit of MMC read operations
- Event viewer
- Setting max connections per share

Connecting to SMB shares by using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for connecting to SMB shares on the IBM Storage Scale cluster.

1. Open the **Shared Folders Microsoft Management Console (MMC)** snap-in:
 - a) Click **Start > Run**.
 - b) Type `fsmgmt.msc` and click **OK**.The **Shared Folders Microsoft Management Console (MMC)** snap-in opens.
2. Connect to the IBM Storage Scale cluster that has the SMB shares:
 - a) Click **Action > Connect to another computer**.
 - b) Type the IP address of the server you want to connect to in the **Another computer** field and click **OK**.

3. In the left pane, click **Shares**.

All SMB shares are listed in the right pane.

Note: If there is a permissions related error when you click **Shares**, verify that you are a member of the local administrators group of the cluster. For more information, see “[Managing SMB shares using MMC](#)” on page 342.

Related tasks

[Creating SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for creating SMB shares on the IBM Storage Scale cluster.

[Modifying or removing SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for modifying or removing SMB shares on the IBM Storage Scale cluster.

[Managing ACLs of SMB shares using MMC](#)

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for managing access control lists (ACLs) of SMB shares on the IBM Storage Scale cluster.

[Modifying offline settings of SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for modifying offline settings of SMB shares on the IBM Storage Scale cluster.

[Viewing active connections to SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing active connections to SMB shares on the IBM Storage Scale cluster.

[Disconnecting active connections to SMB shares using MMC](#)

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for disconnecting active connections to SMB shares on the IBM Storage Scale cluster.

[Viewing open files in SMB shares using MMC](#)

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing open files in SMB shares on the IBM Storage Scale cluster.

[Viewing the number of locks on files in SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing the number of locks on open files in SMB shares on the IBM Storage Scale cluster.

Creating SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for creating SMB shares on the IBM Storage Scale cluster.

1. Open the **Shared Folders Microsoft Management Console (MMC)** snap-in:

a) Click **Start > Run**.

b) Type `fsmgmt.msc` and click **OK**.

The **Shared Folders Microsoft Management Console (MMC)** snap-in opens.

2. Connect to the server on which you want to create SMB shares:

a) Click **Action > Connect to another computer**.

b) Type the IP address of the server you want to connect to in the **Another computer** field and click **OK**.

3. In the left pane, right-click **Shares** and then click **New Share**.

The **Create A Shared Folder** wizard opens.

Note: If there is a permissions related error when you click **Shares**, verify that you are a member of the local administrators group of the cluster. For more information, see “[Managing SMB shares using MMC](#)” on page 342.

4. In the **Create A Shared Folder** wizard, click **Next**.

5. In the **Folder path** field, enter the share path and click **Next**.

Note: The directory for the SMB has to already exist in the file system.

6. Enter the SMB share name and description, select the required offline setting, and then click **Next**.
7. Select the required SMB share permission setting and click **Finish**.

Related tasks

[Connecting to SMB shares by using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for connecting to SMB shares on the IBM Storage Scale cluster.

[Modifying or removing SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for modifying or removing SMB shares on the IBM Storage Scale cluster.

[Managing ACLs of SMB shares using MMC](#)

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for managing access control lists (ACLs) of SMB shares on the IBM Storage Scale cluster.

[Modifying offline settings of SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for modifying offline settings of SMB shares on the IBM Storage Scale cluster.

[Viewing active connections to SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing active connections to SMB shares on the IBM Storage Scale cluster.

[Disconnecting active connections to SMB shares using MMC](#)

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for disconnecting active connections to SMB shares on the IBM Storage Scale cluster.

[Viewing open files in SMB shares using MMC](#)

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing open files in SMB shares on the IBM Storage Scale cluster.

[Viewing the number of locks on files in SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing the number of locks on open files in SMB shares on the IBM Storage Scale cluster.

Modifying or removing SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for modifying or removing SMB shares on the IBM Storage Scale cluster.

1. Open the **Shared Folders Microsoft Management Console (MMC)** snap-in:
 - a) Click **Start > Run**.
 - b) Type `fsmgmt.msc` and click **OK**.The **Shared Folders Microsoft Management Console (MMC)** snap-in opens.
2. Connect to the server on which you want to create SMB shares:
 - a) Click **Action > Connect to another computer**.
 - b) Type the IP address of the server you want to connect to in the **Another computer** field and click **OK**.
3. In the left pane, click **Shares**.
All SMB shares are listed in the right pane.
- Note:** If there is a permissions related error when you click **Shares**, verify that you are a member of the local administrators group of the cluster. For more information, see “[Managing SMB shares using MMC](#)” on page 342.
4. Do one of the following steps depending on whether you want to modify or remove SMB shares:
 - To modify an SMB share:
 - a. In the right pane, right-click the SMB share that you want to modify, and then click **Properties**.

- b. Modify the properties as required and click **OK**.
- To remove an SMB share:
 - a. In the right pane, right-click the SMB share that you want to remove, and then click **Stop Sharing**.

Related tasks

[Connecting to SMB shares by using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for connecting to SMB shares on the IBM Storage Scale cluster.

[Creating SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for creating SMB shares on the IBM Storage Scale cluster.

[Managing ACLs of SMB shares using MMC](#)

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for managing access control lists (ACLs) of SMB shares on the IBM Storage Scale cluster.

[Modifying offline settings of SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for modifying offline settings of SMB shares on the IBM Storage Scale cluster.

[Viewing active connections to SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing active connections to SMB shares on the IBM Storage Scale cluster.

[Disconnecting active connections to SMB shares using MMC](#)

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for disconnecting active connections to SMB shares on the IBM Storage Scale cluster.

[Viewing open files in SMB shares using MMC](#)

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing open files in SMB shares on the IBM Storage Scale cluster.

[Viewing the number of locks on files in SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing the number of locks on open files in SMB shares on the IBM Storage Scale cluster.

Managing ACLs of SMB shares using MMC

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for managing access control lists (ACLs) of SMB shares on the IBM Storage Scale cluster.

1. Open the **Shared Folders Microsoft Management Console (MMC)** snap-in:

- a) Click **Start > Run**.
- b) Type `fsmgmt.msc` and click **OK**.

The **Shared Folders Microsoft Management Console (MMC)** snap-in opens.

2. Connect to the IBM Storage Scale cluster that has the SMB shares:

- a) Click **Action > Connect to another computer**.
- b) Type the IP address of the server you want to connect to in the **Another computer** field and click **OK**.

3. In the left pane, click **Shares**.

All SMB shares are listed in the right pane.

Note: If there is a permissions related error when you click **Shares**, verify that you are a member of the local administrators group of the cluster. For more information, see [“Managing SMB shares using MMC” on page 342](#).

4. In the right pane, right-click the SMB share for which you want to view or change the permissions and then click **Properties**.
5. You can do one of the following:

- To view the permissions a user or a group has for the SMB share, on the **Share Permissions** tab, under the "Group or user names" pane, click on the user name or the group name.
The permissions are displayed in the "Permissions for" pane.
- To change the permissions a user or a group has for the SMB share, on the **Security** tab, under the "Group or user names" pane, click on the user name or the group name and then click **Edit**.

Note: Changes affect only the SMB share, not the ACL in the file system of the exported directory.

For information on permissions that you can change, see documentation for the Shared Folders Microsoft Management Console (MMC) snap-in.

Related tasks

[Connecting to SMB shares by using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for connecting to SMB shares on the IBM Storage Scale cluster.

[Creating SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for creating SMB shares on the IBM Storage Scale cluster.

[Modifying or removing SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for modifying or removing SMB shares on the IBM Storage Scale cluster.

[Modifying offline settings of SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for modifying offline settings of SMB shares on the IBM Storage Scale cluster.

[Viewing active connections to SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing active connections to SMB shares on the IBM Storage Scale cluster.

[Disconnecting active connections to SMB shares using MMC](#)

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for disconnecting active connections to SMB shares on the IBM Storage Scale cluster.

[Viewing open files in SMB shares using MMC](#)

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing open files in SMB shares on the IBM Storage Scale cluster.

[Viewing the number of locks on files in SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing the number of locks on open files in SMB shares on the IBM Storage Scale cluster.

Modifying offline settings of SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for modifying offline settings of SMB shares on the IBM Storage Scale cluster.

1. Open the **Shared Folders Microsoft Management Console (MMC)** snap-in:

- a) Click **Start > Run**.
- b) Type `fsmgmt.msc` and click **OK**.

The **Shared Folders Microsoft Management Console (MMC)** snap-in opens.

2. Connect to the IBM Storage Scale cluster that has the SMB shares:

- a) Click **Action > Connect to another computer**.
- b) Type the IP address of the server you want to connect to in the **Another computer** field and click **OK**.

3. In the left pane, click **Shares**.

All SMB shares are listed in the right pane.

Note: If there is a permissions related error when you click **Shares**, verify that you are a member of the local administrators group of the cluster. For more information, see “[Managing SMB shares using MMC](#)” on page 342.

4. In the right pane, right-click the SMB share whose offline settings you want to modify, and then click **Properties**.
5. On the **General** tab, click **Offline Settings**.
6. In the **Offline Settings** window, configure the offline settings of the SMB share.

For information on offline settings that you can configure, see documentation for the Shared Folders Microsoft Management Console (MMC) snap-in.

Related tasks

[Connecting to SMB shares by using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for connecting to SMB shares on the IBM Storage Scale cluster.

[Creating SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for creating SMB shares on the IBM Storage Scale cluster.

[Modifying or removing SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for modifying or removing SMB shares on the IBM Storage Scale cluster.

[Managing ACLs of SMB shares using MMC](#)

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for managing access control lists (ACLs) of SMB shares on the IBM Storage Scale cluster.

[Viewing active connections to SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing active connections to SMB shares on the IBM Storage Scale cluster.

[Disconnecting active connections to SMB shares using MMC](#)

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for disconnecting active connections to SMB shares on the IBM Storage Scale cluster.

[Viewing open files in SMB shares using MMC](#)

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing open files in SMB shares on the IBM Storage Scale cluster.

[Viewing the number of locks on files in SMB shares using MMC](#)

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing the number of locks on open files in SMB shares on the IBM Storage Scale cluster.

Viewing active connections to SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing active connections to SMB shares on the IBM Storage Scale cluster.

1. Open the **Shared Folders Microsoft Management Console (MMC)** snap-in:

- a) Click **Start > Run**.
- b) Type `fsmgmt.msc` and click **OK**.

The **Shared Folders Microsoft Management Console (MMC)** snap-in opens.

2. Connect to the IBM Storage Scale cluster that has the SMB shares:

- a) Click **Action > Connect to another computer**.
- b) Type the IP address of the server you want to connect to in the **Another computer** field and click **OK**.

3. In the left pane, click **Sessions**.

All active connections to SMB shares are listed in the right pane.

Related tasks

Connecting to SMB shares by using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for connecting to SMB shares on the IBM Storage Scale cluster.

Creating SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for creating SMB shares on the IBM Storage Scale cluster.

Modifying or removing SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for modifying or removing SMB shares on the IBM Storage Scale cluster.

Managing ACLs of SMB shares using MMC

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for managing access control lists (ACLs) of SMB shares on the IBM Storage Scale cluster.

Modifying offline settings of SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for modifying offline settings of SMB shares on the IBM Storage Scale cluster.

Disconnecting active connections to SMB shares using MMC

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for disconnecting active connections to SMB shares on the IBM Storage Scale cluster.

Viewing open files in SMB shares using MMC

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing open files in SMB shares on the IBM Storage Scale cluster.

Viewing the number of locks on files in SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing the number of locks on open files in SMB shares on the IBM Storage Scale cluster.

Disconnecting active connections to SMB shares using MMC

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for disconnecting active connections to SMB shares on the IBM Storage Scale cluster.

1. Open the **Shared Folders Microsoft Management Console (MMC)** snap-in:

- a) Click **Start > Run**.
- b) Type `fsmgmt.msc` and click **OK**.

The **Shared Folders Microsoft Management Console (MMC)** snap-in opens.

2. Connect to the IBM Storage Scale cluster that has the SMB shares:

- a) Click **Action > Connect to another computer**.
- b) Type the IP address of the server you want to connect to in the **Another computer** field and click **OK**.

3. In the left pane, click **Sessions**.

All active connections to SMB shares are listed in the right pane.

4. In the right pane, right-click the connection that you want to close and then click **Close Session**.



Attention: If connections are forced to close, data loss might occur for open files on the connections getting closed.

5. Click **OK** to confirm.

Related tasks

Connecting to SMB shares by using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for connecting to SMB shares on the IBM Storage Scale cluster.

Creating SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for creating SMB shares on the IBM Storage Scale cluster.

Modifying or removing SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for modifying or removing SMB shares on the IBM Storage Scale cluster.

Managing ACLs of SMB shares using MMC

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for managing access control lists (ACLs) of SMB shares on the IBM Storage Scale cluster.

Modifying offline settings of SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for modifying offline settings of SMB shares on the IBM Storage Scale cluster.

Viewing active connections to SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing active connections to SMB shares on the IBM Storage Scale cluster.

Viewing open files in SMB shares using MMC

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing open files in SMB shares on the IBM Storage Scale cluster.

Viewing the number of locks on files in SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing the number of locks on open files in SMB shares on the IBM Storage Scale cluster.

Viewing open files in SMB shares using MMC

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing open files in SMB shares on the IBM Storage Scale cluster.

1. Open the **Shared Folders Microsoft Management Console (MMC)** snap-in:

- a) Click **Start > Run**.
- b) Type `fsmgmt.msc` and click **OK**.

The **Shared Folders Microsoft Management Console (MMC)** snap-in opens.

2. Connect to the IBM Storage Scale cluster that has the SMB shares:

- a) Click **Action > Connect to another computer**.
- b) Type the IP address of the server you want to connect to in the **Another computer** field and click **OK**.

3. In the left pane, click **Open Files**.

All open files in SMB shares are listed in the right pane.

Related tasks

Connecting to SMB shares by using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for connecting to SMB shares on the IBM Storage Scale cluster.

Creating SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for creating SMB shares on the IBM Storage Scale cluster.

Modifying or removing SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for modifying or removing SMB shares on the IBM Storage Scale cluster.

Managing ACLs of SMB shares using MMC

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for managing access control lists (ACLs) of SMB shares on the IBM Storage Scale cluster.

Modifying offline settings of SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for modifying offline settings of SMB shares on the IBM Storage Scale cluster.

Viewing active connections to SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing active connections to SMB shares on the IBM Storage Scale cluster.

Disconnecting active connections to SMB shares using MMC

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for disconnecting active connections to SMB shares on the IBM Storage Scale cluster.

Viewing the number of locks on files in SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing the number of locks on open files in SMB shares on the IBM Storage Scale cluster.

Viewing the number of locks on files in SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing the number of locks on open files in SMB shares on the IBM Storage Scale cluster.

1. Open the **Shared Folders Microsoft Management Console (MMC)** snap-in:

- a) Click **Start > Run**.
- b) Type `fsmgmt.msc` and click **OK**.

The **Shared Folders Microsoft Management Console (MMC)** snap-in opens.

2. Connect to the IBM Storage Scale cluster that has the SMB shares:

- a) Click **Action > Connect to another computer**.
- b) Type the IP address of the server you want to connect to in the **Another computer** field and click **OK**.

3. In the left pane, click **Open Files**.

All open files in SMB shares are listed in the right pane.

4. In the right pane, view locks on a file under the **# Locks** column.

The number of locks is displayed under the **# Locks** column and the type of locks is displayed under the **Open Mode** column.

Related tasks

Connecting to SMB shares by using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for connecting to SMB shares on the IBM Storage Scale cluster.

Creating SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for creating SMB shares on the IBM Storage Scale cluster.

Modifying or removing SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for modifying or removing SMB shares on the IBM Storage Scale cluster.

Managing ACLs of SMB shares using MMC

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for managing access control lists (ACLs) of SMB shares on the IBM Storage Scale cluster.

Modifying offline settings of SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for modifying offline settings of SMB shares on the IBM Storage Scale cluster.

Viewing active connections to SMB shares using MMC

You can use the Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing active connections to SMB shares on the IBM Storage Scale cluster.

Disconnecting active connections to SMB shares using MMC

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for disconnecting active connections to SMB shares on the IBM Storage Scale cluster.

Viewing open files in SMB shares using MMC

You can use Shared Folders Microsoft Management Console (MMC) snap-in on Microsoft Windows clients for viewing open files in SMB shares on the IBM Storage Scale cluster.

Managing NFS exports

Use the following information to manage NFS exports in IBM Storage Scale.

Creating NFS exports

To add an NFS export, use the **mmnfs** export add command.

1. If the directory to be exported does not exist, create the directory by issuing the following commands:

```
mmcrfileset fs01 fileset --inode-space=new  
mmlinkfileset fs01 fileset -J /gpfs/fs01/fileset
```

For more information, see *mmcrfileset command* and *mmlinkfileset command* in *IBM Storage Scale: Command and Programming Reference Guide*.

Note: An independent fileset is recommended for NFS exports.

2. Adjust the ownership and permissions of the folder as required.

Use the GPFS ACLs with **mmgetacl** and **mmpputacl** to set the correct ownership and the access permission.

Additional options can be set during the export creation. For more information about supported options, see *mmnfs command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

3. Create the NFS export by using the following command:

```
mmnfs export add /gpfs/fs01/fileset -c "*(Access_Type=RW)"
```

Note: NFS service restarts after the first export creation.

4. To export the fileset fset_1 to specific set of hosts that fall in the 255.255.0.0 subnet, issue the following command:

```
mmnfs export add /gpfs/fs01/fset_1 --client "10.1.0.0/16(Access_Type=RW)"
```

5. To export the fileset fset_2 to specific set of hosts that fall in the 255.255.255.0 subnet, issue the following command:

```
mmnfs export add /gpfs/fs01/fset_2 --client "10.1.1.0/24(Access_Type=RW)"
```

Note: The directory can be exported with multiple client sections. The sequence of client definitions does matter and the first match determines the effective access.

For more information about client definitions and how to order them, see the *mmnfs command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

GUI navigation

To work with this function in the GUI, log on to the IBM Storage Scale GUI and select **Protocols > NFS Exports**.

Creating an NFS export by using GUI

You can use the NFS client-server communication standard to view, store, and update files on a remote computer. Using the NFS protocol, a client can mount all or a portion of an exported file system from the server to access data. You must configure the NFS server to enable NFS file sharing in the IBM Storage Scale system. The system supports both NFSv3 and NFSv4 versions of the NFS protocol.

You need to enable and start the NFS service on the cluster to create an NFS export in the cluster. You can start and stop the NFS service from the **Services > NFS** page of the IBM Storage Scale GUI.

Perform the following steps to create an NFS export to host or share the data over the NFS protocol. You also need to add NFS clients to access the export. You can add the NFS client either when you create the NFS export or when you edit the export.

1. Go to **Protocols > NFS Exports** page in the IBM Storage Scale GUI. The NFS Exports page appears.
2. Click **Create Export**. The Create Export window appears.
3. Click **Browse** to select the path for the export.
4. Specify the pseudo path of the export in the **Pseudo path** field. This is the path name that is used by the client to locate the directory in the file system. This option is available only when the export is accessed through the NFSv4 protocol.
5. Type the name of the owner of the export in the **Owner** field. This value can either be a user name or a combination of user name and domain name. The default value is `root`. Click **Edit** to change the default value.
6. Type the owning group name in the **Owning group** field. This is the associated group for the path.
7. Click **Add NFS Client** to add NFS clients who can access the export. The Add NFS Client window appears. You can also add an NFS client later by using the **Edit** option that is available in the **Actions** menu.
8. Enter the details of the NFS client in the Add NFS Client window and click **Add**.
9. Click **Create** in the Create Export window to create the export.

An NFS export is created for a path.

Changing NFS export configuration

After an NFS export is created, the export attributes can be changed by using the **mmnfs export change** command.

For the documentation of all supported options, see *mmnfs command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

For example, to grant another client IP address access to the NFS export, run the following command:

```
mmnfs export change /gpfs/fs01/nfs --nfsadd "10.23.23.23(Access_Type=RW)"
```

After the change is made, verify the configuration by running the following command:

```
mmnfs export list
```

The system displays output similar to this:

```
Path Delegations Clients
-----
/gpfs/fs01/nfs none 10.23.23.21
/gpfs/fs01/nfs none 10.23.23.22
/gpfs/fs01/nfs none 10.23.23.23
```

For example, to remove access for a client IP address from the NFS export, run the following command:

```
mmnfs export change /gpfs/fs01/nfs --nfsremove "10.23.23.21"
```

After the change is made, verify the configuration by running the following command:

```
mmnfs export list
```

The system displays output similar to this:

```
Path Delegations Clients
-----
/gpfs/fs01/nfs none 10.23.23.22
/gpfs/fs01/nfs none 10.23.23.23
```

Removing NFS exports

To remove an NFS export, use the **mmnfs export remove** command.

To remove an NFS export, follow these steps:

1. Specify the following command:

```
mmnfs export remove /gpfs/fs01/fset1
```

Here, you want to remove *fset1*. The system displays output similar to the following:

```
The NFS export was deleted successfully.
```

2. Verify that the export is removed by listing the configured NFS exports. Specify:

```
mmnfs export list
```

Listing NFS exports

To list the NFS exports, enter the following command:

```
mmnfs export list
```

The system displays output similar to the following:

```
Path Delegations Clients
-----
/gpfs/FS1/fset_1 NONE *
/gpfs/FS1/fset_2 NONE 10.1.0.0/16
/gpfs/FS1/fset_2 NONE @host_n87_n88
```

Note: You can use **--nfsdefs** or **--nfsdefs-match** as filters with the command. For more information, see the topic *mmnfs command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

GUI navigation for NFS exports

Use the following information to manage NFS exports in IBM Storage Scale.

To work with the NFS exports function in the GUI, log on to the IBM Storage Scale GUI and select **Protocols > NFS Exports**.

Making bulk changes to NFS exports

The **mmnfs export load** command can be used to make bulk changes to existing NFS Exports as an alternative to using the **mmnfs export change** command.

Since the existing command to change an NFS export, **mmnfs export change**, will require a few seconds runtime for every invocation of the command, an alternate method is provided to facilitate bulk changes to the NFS configuration. This procedure can be used, for example, to quickly and easily add additional NFS clients to an export, or to change NFS export attributes, on a per client basis, for any existing NFS client definition.



CAUTION: The `mmnfs export load < NFS_exports_config_file >` command causes a server restart similar to a configuration change. You can use `mmnfs export change` to avoid a server restart.

Existing NFS export

If there is at least one existing NFS export, use the following procedure to make changes to an NFS exports configuration file:

1. Check that there is at least one existing NFS export by issue the following command:

```
mmnfs export list -Y | grep nfsexports | grep -v HEADER
```

2. Check that there is at least one CES node in the cluster:

```
mmces node list -Y | grep -v HEADER
```

3. Check that NFS is enabled:

```
mmces service list -Y | grep NFS:enabled
```

4. Log into a CES node:

```
ssh `mmces node list -Y | grep -v HEADER | tail -1 | awk -F':' '{print $8}'`
```

5. Start NFS (if not already started) on the CES node:

```
mmces service start NFS
```

6. Copy the existing NFS exports configuration file to /tmp:

```
cp -pr /var/mmfs/ces/nfs-config/gpfs.ganesha.exports.conf /tmp/gpfs.ganesha.exports.conf
```

7. Make a backup copy of the original NFS exports configuration file:

```
cp /tmp/gpfs.ganesha.exports.conf /tmp/gpfs.ganesha.exports.conf.bak
```

8. Manually edit /tmp/gpfs.ganesha.exports.conf:

```
vim /tmp/gpfs.ganesha.exports.conf
```

9. When making changes, observe the following guidelines for attributes and values (subject to change at the discretion of the IBM Storage Scale software development team):

```
# EXPORT Only Options (One EXPORT block per Path):
EXPORT {
  Path=<value>; # must be unique
  Pseudo=<value>; # must be unique; usually same as Path
  Tag=<value>; # must be unique; usually same as Path
  Export_id=<value>; # must be unique
  MaxRead=<value>;
  MaxWrite=<value>;
  PrefRead=<value>;
  PrefWrite=<value>;
  PrefReaddir=<value>;
  MaxOffsetWrite=<value>;
  MaxOffsetRead=<value>;
  Filesystem_id=<value>;
  UseCookieVerifier=<value>;
  Attr_Expiration_Time=<value>;
  Delegations=none;
  ...
} # encloses EXPORT block containing FSAL and one or more CLIENT blocks
# EXPORT Option Values:
["PATH"] = "MANDATORY=yes;TYPE=string;DEFAULT=no_default"
["PSEUDO"] = "MANDATORY=yes;TYPE=string;DEFAULT=no_default"
["TAG"] = "MANDATORY=yes;TYPE=string;DEFAULT=no_default"
["EXPORT_ID"] = "MANDATORY=yes;TYPE=value;MIN=1;MAX=65535;DEFAULT=no_default"
```

```

["MAXREAD"] = "MANDATORY=yes;TYPE=value;MIN=512;MAX=67108864;DEFAULT=1048576"
["MAXWRITE"] = "MANDATORY=yes;TYPE=value;MIN=512;MAX=67108864;DEFAULT=1048576"
["PREFREAD"] = "MANDATORY=yes;TYPE=value;MIN=512;MAX=67108864;DEFAULT=1048576"
["PREFWRITE"] = "MANDATORY=yes;TYPE=value;MIN=512;MAX=67108864;DEFAULT=1048576"
["PREFREADDIR"] = "MANDATORY=yes;TYPE=value;MIN=512;MAX=67108864;DEFAULT=1048576"
["MAXOFFSETREAD"] = "MANDATORY=yes;TYPE=value;MIN=512;MAX=18446744073709551615;DEF
AULT=18446744073709551615"
["MAXOFFSETWRITE"] = "MANDATORY=yes;TYPE=value;MIN=512;MAX=18446744073709551615;DE
FAULT=18446744073709551615"
["FILESYSTEM_ID"] = "MANDATORY=yes;TYPE=value;MIN=0;MAX=18446744073709551615;DEFAU
LT=666.666"
["USECOOKIEVERIFIER"] = "MANDATORY=yes;TYPE=bool;DEFAULT=false"
["ATTR_EXPIRATION_TIME"] = "MANDATORY=yes;TYPE=value;MIN=0;MAX=360;DEFAULT=60"
# FSAL Only Options (One FSAL block per EXPORT block):
FSAL {
Name=GPFS;
}
# CLIENT Only Options (One or more CLIENT blocks per EXPORT block):
CLIENT {
Clients=<value>;
Access_Type=<value>;
Protocols=<value>;
Transports=<value>;
Anonymous_uid=<value>;
Anonymous_gid=<value>;
SectType=<value>;
PrivilegedPort=<value>;
Manage_Gids=<value>;
Squash=<value>;
NFS_Commit=<value>;
Delegations=none;
}
# CLIENT Option Values:
["CLIENTS"] = "MANDATORY=yes;TYPE=string;DEFAULT=*&
["ACCESS_TYPE"] = "MANDATORY=yes;TYPE=enum;LIST=none,RW,RO,MDONLY,MDONLY_RO;DEFAU
LT=RO"
["PROTOCOLS"] = "MANDATORY=yes;TYPE=enum;LIST=3,4,NFS3,NFS4,V3,V4,NFSv3,NFSv4;DEFA
ULT=3,4"
["TRANSPORTS"] = "MANDATORY=yes;TYPE=enum;LIST=UDP,TCP;DEFAULT=TCP"
["ANONYMOUS_UID"] = "MANDATORY=yes;TYPE=value;MIN=-
2147483648;MAX=4294967295;DEFAULT=-2"
["ANONYMOUS_GID"] = "MANDATORY=yes;TYPE=value;MIN=-
2147483648;MAX=4294967295;DEFAULT=-2"
["SECTYPE"] = "MANDATORY=yes;TYPE=enum;LIST=none,sys,krb5,krb5i,krb5p;DEFAULT=sys"
["PRIVILEGEDPORT"] = "MANDATORY=yes;TYPE=bool;DEFAULT=false"
["MANAGE_GIDS"] = "MANDATORY=yes;TYPE=bool;DEFAULT=false"
["SQUASH"] = "MANDATORY=yes;TYPE=enum;LIST=root,root_squash,rootsquash,all,all_squ
ash,allsquash,no_root_squash,none,noidsquash;DEFAULT=root_squash"
["NFS_COMMIT"] = "MANDATORY=yes;TYPE=bool;DEFAULT=false"

```

- Load the changes to the NFS exports config file (this will restart NFS on every CES node on which NFS is currently running):

```
mmnfs export load /tmp/gpfs.ganesha.exports.conf
```

Note: The **mmnfs export load** command will conduct a check of the exports configuration file. If the following message is displayed, check the syntax of the NFS exports configuration file, focusing on the changes made in the previous step and try again:

```
mmnfs export load. The syntax of the NFS export configuration file to load is
not correct:
/tmp/gpfs.ganesha.exports.conf.
```

- Verify changes to the NFS configuration via the **mmnfs export list** command:

```
mmnfs export list -Y
```

If a long listing of all NFS exports is desired, use a keyword with the -n option. For example, with `/gpfs` as the keyword (`/gpfs` is the root of each NFS file system in this case):

```
[11:00:48] xxxxx:~:% mmnfs export list -Y -n /gpfs
mmcesnfslsexport:nfsexports:HEADER:version:reserved:reserved:Path:Delegations:Clients:Access_Type:Protocols:Transports:Squash:Anonymous_uid:Anonymous_gid:SecType:PrivilegedPort:DefaultDelegations:Manage_Gids:NFS_Commit:
mmcesnfslsexport:nfsexports:0:1:::/gpfs/fs1/fset1:none:10.0.0.1:R0:3,4:TCP:NO_ROOT_SQUASH:-2:-2:SYS:FALSE:none:FALSE:FALSE:
mmcesnfslsexport:nfsexports:0:1:::/gpfs/fs1/fset1:none:RW:3,4:TCP:ROOT_SQUASH:-2:-2:SYS:FALSE:none:FALSE:FALSE:
```

No existing NFS export

1. Check that there is not an existing NFS export by issuing the following command:

```
mmnfs export list -Y | grep nfsexports | grep -v HEADER
```

2. Create NFS exports (adding the first export restarts NFS on every CES node on which NFS is running. Adding more exports does not restart NFS):

```
mmnfs export add <export>
```

Multiprotocol exports

Exports for SMB and NFSv4 protocols can be configured so that they have access to the same data in the GPFS file system.

To export data by using NFS and SMB, first create an export for one protocol by using the appropriate GPFS command. For example, **mmnfs export add** command. To export the same GPFS path by using a second protocol, create another export by using the protocol-specific export management command. For example, **mmsmb export add** command.

The operations of adding and removing exports do not delete any data in the GPFS file system, and removal of exports does not change the data in the GPFS file system. If later access to a GPFS file system for a specific protocol needs to be removed, this can be done by using the corresponding command. It also does not impact access to the same data configured by using another protocol.

Multiprotocol export considerations

Exports for SMB and NFS protocols can be configured so that they have access to the same data in the file system. In addition, the data can be accessed directly in the file system on the cluster nodes. When configuring access to the same GPFS file system via both the NFS and SMB protocols, certain limitations apply.

These restrictions apply to the general areas of file locking (including share reservation and lock semantics), recovery (reclaim), and cross-protocol notifications.

Access Control Lists (ACLs): In IBM Storage Scale, there is a single common ACL per file or directory in the cluster file system that is used for POSIX, NFS, and SMB access. The SMB server converts each Windows ACL into an NFS4 ACLs for the corresponding file system object.

Shared access (share modes, share reservations): Share modes are feature of the SMB protocol that allows clients to announce what type of parallel access should be allowed by other clients while the file is open. NFSv4 share reservations are the equivalent of SMB share modes for the NFS protocol. There is no equivalent in NFSv3 but IBM Storage Scale allows the SMB server to propagate the share modes into the cluster file system so that NFS clients can honor share modes on commonly used files. The corresponding SMB option is `gpfs:sharemodes`. NFSv4 share reservations are currently not supported.

Note that disabling the SMB option `gpfs:sharemodes` can result in data integrity issues, as SMB application can rely on the enforcement of exclusive access to data to protect the integrity of a file's data. As the SMB file server also does sharemode checks internally, `gpfs:sharemodes` can safely be disabled for data that is only accessed through the SMB protocol.

The important point for POSIX and NFS applications is that file system sharemodes can result in the open() or unlink() system calls to return EACCES. Applications must be prepared to handle this situation.

Details for the interaction of SMB with the file system sharemodes:

When an SMB client requests to open a file, it must specify the allowed share modes. The share modes are specified with the FILE_SHARE_READ, FILE_SHARE_WRITE and FILE_SHARE_DELETE flags; each one of those indicate which parallel access is allowed while the SMB client has the file open.

- If FILE_SHARE_READ is not allowed and another application requests to access the same file for reading data (through the POSIX system call open() with the O_RDONLY flag) then the system call fails with the EACCES error code.
- If FILE_SHARE_WRITE is not allowed and another application requests to access the same file for writing data (through the POSIX system call open() with the O_WRONLY flag), then the system call fails with the EACCES error code.
- If FILE_SHARE_DELETE is not allowed and another application requests to delete the file through the unlink() system call, the system call fails with the EACCES error code.

If another application already has the file open for reading and writing, and the specified share mode from the SMB client conflicts with the existing open, then the SMB client cannot open the file and a "sharing violation" error is returned back to the SMB client.

Share modes in the file system are only enforced if the file is actually opened for READ, EXECUTE, WRITE or APPEND access from the SMB client.

Another limitation of the share mode enforcement in the file system is that it is not possible to grant parallel FILE_SHARE_READ and FILE_SHARE_WRITE access, while not granting FILE_SHARE_DELETE access. In this case, the file system does not enforce that the FILE_SHARE_DELETE restriction and the file can still be deleted.

These limitations only apply to enforcement of sharemodes in the file system. The SMB server also performs internal sharemode checks and handles the sharemode correctly for all SMB access.

Note: The CES NFS server keeps the files accessed by NFSv3 open for a while for performance reasons. This might lead to conflicts during concurrent SMB access to these files. You can use the following command to find out whether the NFS server holds the specified file open:

```
ls /proc/$(pidof gpfs.ganesha.nfssd)/fd -l | grep <file-name>
```

Opportunistic Locking: Optricks are a feature of the SMB protocol that allows clients to cache files locally on the client. If the SMB server is set to propagate oplocks into the cluster file system (gpfs:leases), other clients (NFS, POSIX) can break SMB oplocks. NFS4 delegations are currently not supported.

Byte-range locks: Byte-range locks from SMB clients are propagated into the cluster file system if the SMB option "posix locking" is true. In that case, POSIX and NFS clients are made aware of those locks. Note that for Windows byte-range locks are mandatory whereas for POSIX they are advisory.

File change notifications: SMB clients can request notifications when objects change in the file system. The SMB server notifies its clients about the changes. The notifications include changes that are triggered by POSIX and NFS clients in the directory for which notifications are requested, but not in its subdirectories, if they are done on any CES node. File changes initiated on non-CES cluster nodes will not trigger a notification.

Grace period: The grace period allows NFS clients to reclaim their locks and state for a certain amount of time after a server failure. SMB clients are not aware of NFS grace periods. If you expect a lot of contention between SMB and NFS, NFSv4 reclaims might fail.

Multiprotocol access of protocol exports is only allowed between NFSV4 and SMB. That is, you cannot access the same export by using both NFSV3 and SMB protocols. The reason is that SMB clients typically request exclusive access to a file which does not work with the CES NFS server that keeps files accessed through NFSV3 open.

Chapter 32. Managing S3 protocol

Use the following information to manage S3 services in IBM Storage Scale.

Managing S3 accounts and buckets

The following sections contains information about accounts, objects, and buckets creation.

Managing S3 accounts

Accounts are required to manage S3 user access in IBM Storage Scale. Use the **mms3 account** commands to create, delete, update, and list S3 accounts.

Example

1. Create an S3 account.

```
# mms3 account create s3user9001 --uid 9001 --gid 9000 --newBucketsPath "/gpfs/fs/s3user9001-dir"
```

A sample output is as follows:

```
Account s3user9001 created successfully
Access Key           Secret Key
-----
ei2ISlcY7mvBldlEpa9Z g9LXRBF8bABudHoM0G7YDUBEqLy/XA4tXnXS77vF
```

2. List the created accounts.

```
# mms3 account list
```

A sample output is as follows:

Name	New Buckets Path	Uid	Gid	User
s3user9001	/gpfs/fs/s3user9001-dir	9001	9000	None

The S3 access keys generated in the preceding step can be used by S3 applications to submit authenticated S3 requests to the S3 service.

Managing S3 buckets using AWS CLI

The S3 APIs for managing buckets can be used to manage the S3 buckets at the S3 client. The S3 command of the AWS command line interface (CLI) is used in the following step to create buckets. An alias is created for the AWS CLI that uses the S3 access keys and the CES IP of the CES node where S3 is enabled and running. This procedure is an example of managing S3 buckets. The listing of buckets and objects does not show any results, because no buckets or objects are created.

Example

1. Create an alias on the application node.

```
alias s3u9k='AWS_ACCESS_KEY_ID=ei2ISlcY7mvBldlEpa9Z AWS_SECRET_ACCESS_KEY=g9LXRBF8bABudHoM0G7YDUBEqLy/XA4tXnXS77vF aws --endpoint https://172.20.100.33:6443 --no-verify-ssl s3'
```

Note: Here the endpoint URL has the CES IP, which is shared by the system administrator with all the application users. In cluster environment, the DNS is configured and the CES IP address or name are directed appropriately to the S3 service, which is running on the protocol node.

2. List the buckets , if any.

```
s3u9k ls
```

3. Create a bucket by using the AWS CLI command. The bucket is created.

```
s3u9k mb s3://newbucket-9kuser
```

A sample output is as follows:

```
make_bucket: newbucket-9kuser
```

4. List the bucket by using the AWS CLI command. The bucket is listed.

```
s3u9k ls
```

A sample output is as follows:

```
2024-07-03 05:38:20 newbucket-9kuser
```

5. Upload the objects to the respective bucket.

```
s3u9k cp /root/file_1M s3://newbucket-9kuser
```

A sample output is as follows:

```
upload: ./file_1M to s3://newbucket-9kuser/file_1M
```

6. Download the objects from the bucket.

```
s3u9k cp s3://newbucket-9kuser/file_1M file_2M
```

A sample output is as follows:

```
download: s3://newbucket-9kuser/file_1M to file_2M
```

7. List the content of the S3 bucket. The uploaded file is listed.

```
s3u9k ls s3://newbucket-9kuser
```

A sample output is as follows:

```
2024-07-03 05:51:47 1048576 file_1M
```

The S3 user is unaware of where the object is stored in the underlying filesystem. The S3 protocol creates the buckets under the NewBucketsPath mentioned while creating the S3 user account.

Managing S3 buckets using the mms3 command

Use the **mms3 bucket** commands to create, delete, update, and list S3 buckets.

Note: S3 bucket can be also be created by an S3 account by using the S3 CreateBucket API.

Example: Create a bucket for an existing S3 user account.

1. List the secret and access keys for the user account.

```
# mms3 account list s3user9001
```

Name	New Buckets Path	Uid	Gid	Access Key	Secret Key
s3user9001	/gpfs/fs/s3user9001-dir	9001	9000	ei2IS1cY7mvBldlEpa9Zg9LXRBF8bABudHoM0G7YDUBEqLy/XA4tXnXS77vF	

2. Create a bucket in the directory.

```
# mms3 bucket create newbucket-9k-mms3 --accountName s3user9001 --filesystemPath /gpfs/fs/s3user9001-dir/newbucket-9k-mms3-dir
```

A sample output is as follows:

```
Bucket newbucket-9k-mms3 created successfully.
```

3. List the bucket.

```
# mms3 bucket list newbucket-9k-mms3
```

A sample output is as follows:

Name	Filesystem Path	Bucket Owner
newbucket-9k-mms3	/gpfs/fs/s3user9001-dir/newbucket-9k-mms3-dir	s3user9001

4. At the S3 client the S3 account user can list the buckets by using the AWS commands to view the bucket.

W

```
s3u9k ls
2024-07-03 05:38:20 newbucket-9kuser
2024-07-03 06:26:59 newbucket-9k-mms3
```

Note: s3u9k is the alias that is created earlier on the application node.

5. Upload objects to the created bucket.

```
s3u9k cp /bin/trust s3://newbucket-9k-mms3
upload: ../bin/trust to s3://newbucket-9k-mms3/trust
```

6. List the bucket content by using the AWS CLI command.

```
s3u9k ls s3://newbucket-9k-mms3
2024-07-03 06:35:48 239200 trust
```

Data share across S3 users via bucket policies. S3users can share data across them by using a fine granularity with the help of a having a bucket policy in place. For more information, see <https://community.ibm.com/community/user/storage/blogs/ravi-kumar-komanduri/2024/07/26/ibm-storage-scale-ces-s3-dashshare-bucket-policy>.

S3 objects API

IBM Storage Scale allows S3 users to manage objects by using the following API requests:

- S3 PutObject
- S3 GetObject
- S3 HeadObject
- S3 CopyObject
- S3 DeleteObject
- S3 DeleteObjects
- S3 CreateMultipartUpload
- S3 CompleteMultipartUpload
- S3 AbortMultipartUpload
- S3 UploadPart
- S3 UploadPartCopy
- S3 ListParts
- S3 DeleteObjectTagging
- S3 GetObjectTagging
- S3 PutObjectTagging

- S3 GetBucketPolicy
- S3 SetBucketPolicy
- S3 GetBucketVersioning

IBM Storage Scale S3 allows S3 applications to store user-defined object metadata in addition to the object data itself.

S3 buckets API

IBM Storage Scale S3 maps each S3 bucket to a directory in the IBM Storage Scale file system. IBM Storage Scale S3 allows S3 clients to manage S3 buckets by using the following API requests:

- S3 CreateBucket
- S3 ListObjects
- S3 ListObjectsV2
- S3 DeleteBucket
- S3 HeadBucket
- S3 ListBuckets
- S3 ListMultipartUploads

Other

- Bucket policies
- Support MD5 based ETags
- Upload objects with pre-signed URLs

Backing up the S3 configuration data

The S3 configuration data can be backed up automatically or manually.

Automatic backup of the S3 configuration data

IBM Storage Scale S3 service supports periodic or automatic backup of the S3 configuration data from CES share root directory to the clustered configuration repository (CCR) periodically. On CES nodes where S3 is enabled, the configuration data is backed up every 10 minutes . The S3 configuration data is backed up only on one CES node at a time.

1. To list a backup file of the tar.bz2 S3 configuration on the CCR, issue the following command:

```
# mmccr flist | grep _s3
```

A sample output is as follows:

```
37 _s3-config-backup.tar.bz2
```

2. To get a local copy of the tar.bz2 backup file from the CCR, issue the following command:

```
# mmccr fget _s3-config-backup.tar.bz2 /tmp/s3-config.tar.bz2
```

A sample output is as follows:

```
fget:37
```

3. To list the information of the local backup file, issue the following command:

```
# ls -ltr /tmp/s3-config.tar.bz2
```

A sample output is as follows:

```
-rw----- 1 root root 537 Mar 18 20:34 /tmp/s3-config.tar.bz2
```

Manual backup of the S3 configuration data

The S3 configuration can be backed up by using the **mms3 config backup** command. The **mms3 config backup** command archives all the files and directories at the `<cesSharedRoot path>/ces/s3-config` directory, into a single `.tar.bz2` file, and saves the file to the specified `backup_directory` path. For more information, see the *mms3 command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

1. To know the syntax of the **mms3 config backup** command, run the following command:

```
# mms3 config backup --help
```

The information about the **mms3** command is as follows:

```
Backup the IBM Storage Scale S3 configuration.

Usage:
  mms3 config backup <backup_directory> [flags]

Flags:
  -h, --help    help for backup
Global Flags:
  -d, --debug   verbose logging
```

2. To back up the S3 configuration data manually, issue the below command:

Example,

```
# mms3 config backup /tmp/demo-backup
```

The sample output is as follows:

```
Backup of IBM Storage Scale S3 configuration <backup_directory>/
s3ConfigBackup.20240709-063551.tar.bz2 successful.
```

After the S3 configuration data is backed up successfully, a file name, such as `s3ConfigBackup.date-time.tar.bz2`, is created in the specified `backup_directory`.

Restoring the S3 configuration data

IBM Storage Scale S3 service support restore of the S3 configuration data by using the **mms3 config restore** command. The **mms3 config restore** command extracts the specified `tar.bz2` backup file, overwrites the existing content of `{cesSharedRoot_dir}/ces/mms3-config.json` file, `{cesSharedRoot_dir}/ces/s3-config` directory, and restarts the S3 service on all the CES nodes.

The **mms3 config restore** command details are as follows:

```
# mms3 config restore -h
Restore the IBM Storage Scale S3 configuration.

Usage:
  mms3 config restore <backup_tar_file> [flags]

Flags:
  --force    Allows the config restore action to go through without confirmation from user.
  -h, --help   help for restore

Global Flags:
  -d, --debug  verbose logging
```

For more information, see the *mms3* command in the *IBM Storage Scale: Command and Programming Reference Guide*

Restoring the S3 configuration from the manual or automatic tar.bz2 backup file

The S3 configuration data can be restored to the CES shared root manually or automatically from the *tar.bz2* backup file in two ways:

- Restore the S3 configuration data by using the *tar.bz2* backup file of the S3 configuration, which was automatically backed up in the CCR.

1. To list the *tar.bz2* backup files of S3 configuration in the CCR, issue the following command:

```
# mmccr flist | grep _s3
```

A sample output is as follows:

```
version  name
-----
1       ccr.nodes
1       ccr.disks
1       mmLockFileDB
1       genKeyData
2       genKeyDataNew
35      mmsdrfs
1       mmsysmon.json
4       _callhomeconfig
2       _perfmon.keys
1       measurements.json
1       zmrules.json
3       collectors
6       _gui.settings
15      _gui.user.repo
14      _gui.keystore_settings
31      cesiplist
3       gpfs.ganesha.main.conf
1       gpfs.ganesha.nfsd.conf
1       gpfs.ganesha.log.conf
2       gpfs.ganesha.exports.conf
1       gpfs.ganesha.statdargs.conf
1       idmapd.conf
808     _s3-config-backup.tar.bz2
1       _ces_s3.master_keys
1       authccr
```

2. To get a local copy of the backup *tar.bz2* file from the CCR, issue the following command:

```
# mmccr fget _s3-config-backup.tar.bz2 /tmp/_s3-config-backup.tar.bz2
```

A sample output is as follows:

```
fget:808
```

To extract the directory, issue the following command:

```
# tar -xvf /tmp/_s3-config-backup.tar.bz2 -C /tmp/backup/
```

A sample output is as follows:

```
./
./config.json
./accounts/
./accounts/demo-account.json
./buckets/
./buckets/demo-bucket.json
./system.json
./access_keys/
./access_keys/Pei00g7t1MD4mFfq5qi5.symlink
mms3-config.json
_ces_s3.master_keys
```

3. Ensure that the extracted directory contains `system.json` and `config.json` files, and the `_ces_s3.master_keys` key.

```
# tree /tmp/backup/
```

A sample output is as follows:

```
└── access_keys  
└── accounts  
└── buckets  
└── config.json  
└── mms3-config.json  
└── _ces_s3.master_keys  
└── system.json
```

4. To restore the S3 configuration, issue the following command:

```
# mms3 config restore /tmp/_s3-config-backup.tar.bz2
```

A sample output is as follows:

```
The S3 configuration directory at path /mnt/cesSharedRoot/ces/s3-config is not empty.  
Restore operation will overwrite the existing S3 configuration and restart the S3 service  
on all CES nodes. Do you want to proceed ? [y|n]:y  
Started Restoring s3-config configuration data  
  
Restore of IBM Storage Scale S3 configuration successfully done.  
  
S3 Configuration successfully changed. S3 service restarted on the CES node:  
ces-12.openstacklocal.  
  
S3 Configuration successfully changed. S3 service restarted on the CES node:  
ces-13.openstacklocal.  
  
S3 Configuration successfully changed. S3 service restarted on the CES node:  
ces-14.openstacklocal.
```

- Restore the S3 configuration by using the `.tar.bz2` backup file of the S3 configuration, which you backed up manually by using the **mms3 config backup** command.

To restore the S3 configuration, issue the following command:

```
# mms3 config restore /tmp/s3ConfigBackup.<date>-<time>.tar.bz2
```

```
The S3 configuration directory at path /mnt/cesSharedRoot/ces/s3-config is not empty. Restore  
operation will overwrite the existing S3 configuration and restart the S3 service on all CES  
nodes. Do you want to proceed ? [y|n]:y  
  
Started Restoring s3-config configuration data  
  
Restore of IBM Storage Scale S3 configuration successfully done.  
  
S3 Configuration successfully changed. S3 service restarted on the CES node:  
ces-12.openstacklocal.  
  
S3 Configuration successfully changed. S3 service restarted on the CES node:  
ces-13.openstacklocal.  
  
S3 Configuration successfully changed. S3 service restarted on the CES node:  
ces-14.openstacklocal.
```


Chapter 33. Managing Swift Object storage

Important:

- CES Swift Object protocol feature is not supported from IBM Storage Scale 5.2.0 onwards.
- IBM Storage Scale 5.1.8 is the last release that has CES Swift Object protocol.
- IBM Storage Scale 5.2.0 will tolerate the update of a CES node from IBM Storage Scale 5.1.8.
 - *Tolerate* means:
 - The CES node will be updated to 5.2.0.
 - Swift Object support will not be updated as part of the 5.2.0 update.
 - You may continue to use the version of Swift Object protocol that was provided in IBM Storage Scale 5.1.8 on the CES 5.2.0 node.
 - IBM will provide usage and known defect support for the version of Swift Object that was provided in IBM Storage Scale 5.1.8 until you migrate to a supported object solution that IBM Storage Scale provides.
 - CES Swift Object is replaced with IBM Storage Scale S3. For more details, refer the *S3 support overview* section the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
 - For more information about Swift Object in IBM Storage Scale, refer to the [IBM Storage Scale 5.2.0 documentation](#).
- Please contact IBM for further details and migration planning.

Understanding and managing Object services

Use the following information to manage services that are related to IBM Storage Scale for Object storage.

You can use the **mmces service** command to enable, start, stop, or disable Object services on all protocol nodes.

The enable and disable operations are cluster-wide operations. To enable or disable the Object protocol, use **mmces service [enable | disable] OBJ**. The Object protocol must be initially configured by using the **mmobj swift base** command before it can be enabled in the cluster.



CAUTION: Disabling the Object service unconfigures the object protocol and discards OpenStack Swift configuration and ring files from the CES cluster. If Openstack Keystone configuration is configured locally, disabling Object storage also discards the Keystone configuration and database files from the CES cluster. However, to avoid accidental data loss, the associated filesets that are used for the object data are not automatically removed during disable. The filesets for the object data and any filesets that are created for optional Object storage policies need to be removed manually. If you plan to re-enable the object protocol after you disable it, you must access the repository during the object disablement to reset to the original default object configuration. After, for enabling the object service, either different fileset names need to be specified or the existing filesets need to be cleaned up. For information on cleaning up the object filesets, see the steps *"Remove the fileset that is created for object"* and *"Remove any fileset that is created for an object storage policy"*(if applicable) in the *Cleanup procedures that are required if reinstalling with the spectrumscale installation toolkit* topic of *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Note: To disable the object protocol, first remove the object authentication. For complete usage information, see the **mmuserauth** command in the *IBM Storage Scale: Command and Programming Reference Guide*.

In addition, enabled Object service can be started and stopped on individual nodes or cluster-wide. To start or stop the object protocol cluster-wide, use the following command:

```
mmces service [start | stop] OBJ -a
```

To start or stop the object protocol on individual nodes, use the following command:

```
mmces service [start | stop] OBJ -N <node>
```



Attention: If Object services on a protocol node are stopped by the administrator manually, access to object data might be impacted unless the CES IP addresses are first moved to another node. You can accomplish this in multiple ways, but the simplest is to suspend the node. After you suspend a node, CES automatically moves the CES IPs to the remaining nodes in the cluster. However, doing this suspends operation for all protocols that are running on that protocol node.

If you want to stop object services on a protocol node, you can use the following steps:

1. Suspend CES operations on the protocol node by using the **mmces node suspend** command.
2. View the CES IP addresses on that node by using the **mmces address list** command and verify that all CES IP addresses are moved to other protocol nodes.
3. Stop the object services by using the **mmces service stop OBJ** command.

Note: All Object services must be controlled by using only the **mmces service start/stop** and **systemctl** commands. The explicit use of system or Swift commands to manage services, like **swift-init** or **systemctl**, is not supported and might cause your system to operate incorrectly.

Performing these steps ensures that Object functionality is available on other nodes in the cluster.

To restore Object services on that protocol node, you can use the following steps:

1. Resume CES operations on the protocol node by using the **mmces node resume** command.
2. View the CES IP addresses on that node by using the **mmces address list** command and verify that all CES IP addresses are moved to that protocol node.
3. Start the Object services by using the **mmces service start OBJ** command.

Use the **mces service list** command to list the protocols enabled on IBM Storage Scale. List a verbose output of Object services that are running on the local node by using the **-v** flag as shown in the following example:

```
# mmces service list -v
Enabled services: OBJ SMB NFS
OBJ is running
  OBJ:openstack-swift-object-updater      is running
  OBJ:openstack-swift-object-expirer      is running
  OBJ:ibmobjectizer                     is running
  OBJ:openstack-swift-object-auditor     is running
  OBJ:openstack-swift-object             is running
  OBJ:openstack-swift-account            is running
  OBJ:openstack-swift-container          is running
  OBJ:memcached                         is running
  OBJ:openstack-swift-proxy              is running
  OBJ:openstack-swift-object-replicator   is running
  OBJ:openstack-swift-account-reaper     is running
  OBJ:openstack-swift-account-auditor    is running
  OBJ:openstack-swift-container-auditor   is running
  OBJ:openstack-swift-container-updater   is running
  OBJ:openstack-swift-account-replicator   is running
  OBJ:openstack-swift-container-replicator  is running
  OBJ:openstack-swift-object-sof          is running
  OBJ:postgresql-obj                   is running
  OBJ:httpd (keystone)                  is running
SMB is running
NFS is running
```

For complete usage information, see *mmces command* in *IBM Storage Scale: Command and Programming Reference Guide*.

Every Object protocol node can access every virtual device in the shared file system, and some OpenStack Swift object services can be optimized to take advantage of this by running from a single Object protocol node.

Even though objects are not replicated by OpenStack Swift, the **swift-object-replicator** runs to periodically clean up tombstone files from deleted objects. It is run on a single Object protocol node and manages cleanup for all of the virtual devices.

The **swift-object-updater** is responsible for updating container listings with objects that were not successfully added to the container when they were initially created, updated, or deleted. Like the object replicator, it is run on a single object protocol node.

The following table shows each of the Object services and the set of Object protocol nodes on which they need to be run.

<i>Table 23. Object services and protocol nodes</i>	
Object service	GPFS protocol node
ibmobjectizer	object_singleton_node ¹
openstack-swift-account	All
openstack-swift-account-auditor	object_singleton_node
openstack-swift-account-reaper	All
openstack-swift-account-replicator	All
openstack-swift-container	All
openstack-swift-container-auditor	object_singleton_node
openstack-swift-container-updater	object_singleton_node
openstack-swift-container-replicator	All
openstack-swift-object	All
openstack-swift-object-auditor	object_singleton_node ²
openstack-swift-object-replicator	All
openstack-swift-object-sof	All ¹
openstack-swift-object-updater	object_singleton_node
openstack-swift-object-expirer	object_singleton_node
openstack-swift-proxy	All
memcached	All
httpd (RHEL) or apache2 (Ubuntu)	All ³
postgresql-obj	object_database_node ³

¹ If unified file and object access is enabled.
² If multi-region object deployment is enabled.
³ If local OpenStack Keystone Identity Service is configured.

Understanding the mapping of OpenStack commands to IBM Storage Scale administrator commands

Use this information to map OpenStack commands to IBM Storage Scale administrator commands.

In IBM Storage Scale, for Object storage, several OpenStack commands are replaced with IBM Storage Scale commands for better maintenance. The following information identifies those commands.

1. Ring building

The `swift-ring-builder` command must be used to view the object, container, and account ring on any IBM Storage Scale protocol node.

Note: You must not directly run any commands that modify the ring.

All ring maintenance operations are handled automatically by the CES infrastructure.

For example, when a new CES IP address is added to the configuration, all rings are automatically updated to distribute Swift virtual devices evenly across CES IP addresses.

The controller copy of each ring builder file is kept in the IBM Storage Scale Cluster Configuration Repository (CCR). Changes made locally to the ring files are overwritten with the controller copy when monitoring detects a difference between the ring file in CCR and the file in `/etc/swift`.

2. Configuration changes

The `openstack-config` command must not be used to update any of the configuration files that are consumed by IBM Storage Scale for Object storage. Furthermore, you must not edit these files directly, but instead modify them using the `mmobj config change` command.

The controller copy of object and related configuration files are kept in the IBM Storage Scale CCR. Changes made locally to these config files are overwritten with the controller copy when monitoring detects a difference between the configuration file in CCR and the file in `/etc/swift` or `/etc/keystone`.

Changing Object configuration values

Use the following information to change the Object configuration values in the Cluster Configuration Repository (CCR).

You can manage the Object configuration data in the Cluster Configuration Repository (CCR). When an Object configuration changes, callbacks on each protocol node update that node with the change and restart one or more Object services if necessary.

To change the Object configuration, use the `mmobj` command so the change is made in the CCR and propagated correctly across the Swift cluster.

For more information, see the *mmobj command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

How to change the object base configuration to enable S3 API

IBM Storage Scale uses the s3api middleware for OpenStack Swift, which allows access to IBM Storage Scale by using Amazon Simple Storage Service (S3) API. Swift S3 serves as a linkage between legacy S3 applications and Swift storage. Not all S3 features are supported. The S3 bucket is mapped to a Swift container, which means that a bucket has the same limitations as a container. For more information about compatibility with AWS S3, see [S3/Swift REST API Comparison Matrix](#) in OpenStack documentation.

Use the following steps if S3 API is not enabled as part of the object base configuration:

1. To enable S3 API, run the following command:

```
mmobj s3 enable
```

2. To verify that S3 API is enabled, run the following command:

```
mmobj s3 list
```

3. To disable S3 API, run the following command:

```
mmobj s3 disable
```

4. To verify that S3 API is disabled, run the following command:

```
mmobj s3 list
```

Remember: You can use the s3api middleware for OpenStack Swift with S3 clients that are using either the V2 or V4 S3 protocol.

The V2 protocol is the default. If you use the V4 protocol, make sure that the region of the request matches the value of the location property in the filter:s3api section of proxy-server.conf file.

Note: The default value for location in the s3api middleware is us-east-1, which means that V4 S3 clients must set us-east-1 as the region.

You can change the location value to something other than us-east-1 by changing the property in the proxy-server.conf file. To change the location, run the following command:

```
mmobj config change --ccrfile "proxy-server.conf" --section "filter:s3api" --property "location" --value "NEW_LOCATION"
```

Replace "NEW_LOCATION" with the appropriate value for your environment.

Remember: After you change the value, any S3 clients that are using the V4 protocol must set their region to the same value.

For the listing of buckets with the S3 protocol, a hardcoded date is returned as the creation date of each bucket because of a limitation in Swift. For example, the date might look like this:

```
2009-02-03 10:45:09
```

Important: To get the actual creation date of the bucket, use the Swift protocol to query the associated container instead.

Configuring OpenStack EC2 credentials

The credentials that the Amazon S3 and Elastic Compute Cloud (EC2) APIs use are different from the credentials that OpenStack API uses. So, you must generate these special credentials to use them when you access the IBM Storage Scale OpenStack services.

The credentials are created by the **openstack** command, a command-line client for OpenStack that allows the creation and use of access or secret pairs for a user or project pair. When you use the command, you must create the access or secret for each user or project pair:

1. Source openrc with the administrative credentials.
2. Create EC2 credential by running this command for user-defined blob as a credential:

```
openstack credential create --type ec2 --project <project> <user> '{"access": "<aws_access_key>", "secret": "<aws_secret_key>"}'
```

Note: Make sure that you use Keystone UUIDs rather than names if duplicate user or project names might exist across domains. Additionally, the administrative users must be able to list and delete access or secrets for a specific user or project.

You can set <aws_access_key> and <aws_secret_key> to any value. These values are supplied to the S3 client. These values are typically set as the access and secret S3 values. S3 uses them when it connects to Object storage. The S3 layer in OpenStack uses these values to look up the associated user and project that is associated with the EC2 credential.

3. View all EC2 credentials by running this command:

```
openstack credential list  
openstack credential show <credential-id>
```

4. You can change your Access Key ID and Secret Access Key if necessary.

Note: You might want to consider a regular rotation of these keys and switching applications to use the new pair.

Change the EC2 credentials by running this command:

```
openstack credential set --type ec2 --data '{"access": <access>, "secret": <secret>}' --project <project> <credential-id>
```

5. Delete the EC2 credentials by running this command:

```
openstack credential delete <credential_id>
```

The following example shows the creation of EC2 credentials that link with the S3 credentials "s3user" and "s3pass" to the Keystone user "admin" that is in the project "build":

```
source /root/openrc
openstack credential create --type ec2 --project build admin '{"access": "s3user", "secret": "s3pass"}'
```

Now you can connect to the IBM Storage Scale Object store by using the Amazon S3 API. You can connect with any S3-enabled client by using the access key "s3user" and the secret "s3pass".

How to manage the OpenStack S3 API

The following topic lists the permissions and the known limitations of S3 API.

IBM Storage Scale supports S3 access control lists (ACLs) for buckets and objects. These S3 ACLs are stored separately from the ACLs that are set through the Swift API or the ACLs stored in the file system (NFSv4 or POSIX).

You can set and query ACLs through S3 API. For more information, see the [Amazon S3 documentation](#).

If the S3 API is enabled, the default value of **s3_acl** in the `proxy-server.conf` file is `true`. The S3 API uses its own metadata for an ACL. The metadata includes the X-Container-Sysmeta-Swift3-Acl, which is used to achieve the best S3 compatibility.

However, if the S3 API is set to `false`, the S3 API initially uses Swift ACLs (such as the X-Container-Read ACL) initially instead of S3 ACLs.

To use the S3 API in IBM Storage Scale, you must have a role that is defined for the swift project. Any role suffices because for the S3 API there is no difference between the `SwiftOperator` role or other roles.

The owner of a resource is implicitly granted **FULL_CONTROL** instead of just **READ_ACP** and **WRITE_ACP**. Granting this control is safe (not a security issue) because with **WRITE_ACP**, the owners can grant themselves **FULL_CONTROL** access.

The following table lists the required permissions for S3 operations.

S3 operation	Required permission
PUT object	WRITE permission on bucket or as bucket owner is required.
HEAD object	READ permission on object or as object owner is required.
GET object	READ permission on object or as object owner is required.
DELETE object	WRITE permission on bucket or as bucket owner is required.
Get object ACL (GET on ACL subresource)	READ_ACP permission on object or as object owner is required.
Set object ACL (PUT on ACL subresource)	WRITE_ACP permission on object or as object owner is required.
Create bucket (PUT)	Any user with a role on the project can create a bucket.
HEAD bucket	READ permission on bucket or as bucket owner is required.

S3 operation	Required permission
GET bucket	READ permission on bucket or as bucket owner is required.
DELETE bucket	It must be the bucket owner.
Get bucket ACL (GET on ACL subresource)	READ_ACP permission on bucket or as bucket owner is required.
Set bucket ACL (PUT on ACL subresource)	WRITE_ACP permission on bucket or as bucket owner is required.

Known limitations for S3 API support

Known limitations for S3 API support include the following situations:

- The OpenStack Swift S3 API implements a limited set of the functions that are provided by the Amazon S3 API. For more information, see the [OpenStack S3 compatibility matrix](#).
- The OpenStack Swift S3 API maps S3 buckets to Swift containers. High transaction throughput to a S3 bucket might experience performance issues because of container limitations. To avoid these performance issues, spread the requests among many buckets to avoid the underlying containers from being overloaded.
- Unauthorized S3 requests are not supported. S3 requests do not contain a reference to the account, and the object server derives the account information from the authorization information (which is not possible for unauthorized requests).
- You cannot specify S3 ACL grantees by email.
- Grantees in the ACL are not validated. So, any name can be used, including names for users that do not exist.
- The S3 ACLs are not supported in the Objects page of the IBM Storage Scale GUI.
- Container or objects that are created by using the Swift API are not accessible through S3 API when the `allow_no_owner` configuration flag is set to `false` in the `proxy-server.conf` file. To change this setting, you use the following command:

```
mmobj config_change --ccrfile proxy-server.conf --section filter:s3api
--property allow_no_owner --value true
```

The default value of the `allow_no_owner` configuration flag is `true`.

- The POST operation to update metadata is not implemented.

Managing object capabilities

You can manage the object capabilities by using the following commands.

Important:

- CES Swift Object protocol feature is not supported from IBM Storage Scale 5.2.0 onwards.
- IBM Storage Scale 5.1.8 is the last release that has CES Swift Object protocol.
- IBM Storage Scale 5.2.0 will tolerate the update of a CES node from IBM Storage Scale 5.1.8.
 - Tolerate* means:
 - The CES node will be updated to 5.2.0.
 - Swift Object support will not be updated as part of the 5.2.0 update.
 - You may continue to use the version of Swift Object protocol that was provided in IBM Storage Scale 5.1.8 on the CES 5.2.0 node.

- IBM will provide usage and known defect support for the version of Swift Object that was provided in IBM Storage Scale 5.1.8 until you migrate to a supported object solution that IBM Storage Scale provides.
- CES Swift Object is replaced with IBM Storage Scale S3. For more details, refer the *S3 support overview* section the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
- For more information about Swift Object in IBM Storage Scale, refer to the [IBM Storage Scale 5.2.0 documentation](#).
- Please contact IBM for further details and migration planning.
- To list all object capabilities available cluster wide, use the **mmobj config list** command as follows:

```
mmobj config list --ccrfile spectrum-scale-object.conf --section capabilities
```

The system displays output similar to this output:

```
file-access-enabled: true
multi-region-enabled: true
s3-enabled: false
```

You can also list specific object capabilities by using the **mmobj config list** command as follows:

```
mmobj config list --ccrfile spectrum-scale-object.conf --section capabilities
--property file-access-enabled
mmobj config list --ccrfile spectrum-scale-object.conf --section capabilities
--property multi-region-enabled
mmobj config list --ccrfile spectrum-scale-object.conf --section capabilities
--property s3-enabled
```

- Use the **mmobj** command to enable object capabilities:
 - To enable or disable the file-access object capability, use the following commands:
 - Use the **mmobj file-access enable** command to enable file-access capability.
 - Use the **mmobj file-access disable** command to disable file-access capability.
 - To enable or disable the multiregion object capability, use the following commands:
 - Use the **mmobj multiregion enable** command to enable multiregion capability.
 - Use the **mmobj multiregion disable** command to disable multiregion capability.
 - To enable or disable the S3 object capability, use the following commands:
 - Use the **mmobj s3 enable** command to enable S3 object capability.
 - Use the **mmobj s3 disable** command to disable S3 object capability.

Managing object versioning

Use the following topics to enable and disable object versioning.

Enabling object versioning

To enable object versioning, use the following steps.

Note: Object versioning is only supported with objects stored in traditional Swift object storage format. Containers which store their objects in unified file and object storage policies cannot be used with object versioning.

1. Add the section `filter:versioned_writes` to the `proxy-server.conf` file:

- a) To determine whether the section is present in the file, run the following command:

```
mmobj config list --ccrfile proxy-server.conf --section "filter:versioned_writes"
```

The command displays output similar to this output when the section is present:

```
[filter:versioned_writes]
use = egg:swift#versioned_writes
```

- b) If the section is not present, run the following command to add it:

```
mmobj config change --ccrfile proxy-server.conf --section "filter:versioned_writes" --property "use" --value "egg:swift#versioned_writes"
```

2. To set the `filter:versioned_writes` attribute to true, run the following command:

```
mmobj config change --ccrfile proxy-server.conf --section "filter:versioned_writes" --property "allow_versioned_writes" --value "true"
```

3. Add the `versioned_writes` module to the proxy-server pipeline:

- a) To determine whether the module is present in the pipeline, run the following command:

```
mmobj config list --ccrfile proxy-server.conf --section "pipeline:main" --property "pipeline"
```

The command displays the pipeline module list as in the following example:

```
pipeline = healthcheck cache . . . slo dlo versioned_writes proxy-server
```

- b) If the `versioned_writes` module is not included in the pipeline module list, add it to the pipeline immediately before the `proxy-server` module.

To add it to the pipeline, run the `mmobj` command. Make sure that the command is all on one line when you enter it:

```
mmobj config change --ccrfile proxy-server.conf --section "pipeline:main" --property "pipeline" --value "healthcheck
cache . . . slo dlo versioned_writes proxy-server"
```

This command is an example. When you run the command on your system, follow these steps:

- i) In the `--value` parameter, specify the actual list of pipeline modules that are displayed on your system in the output when you determine whether the module is present in the pipeline. Enclose the list in double quotation marks.
- ii) In the list of pipeline modules that you specify, insert the `versioned_writes` pipeline module immediately before the `proxy-server` module.

Disabling object versioning

Use the following steps to disable object versioning.

To disable object versioning across the cluster, run the following command:

```
# mmobj config change --ccrfile proxy-server.conf --section 'filter:versioned_writes' --property allow_versioned_writes --value
false
```

A sample output is as follows:

```
[filter:versioned_writes]
use = egg:swift#versioned_writes
allow_versioned_writes = false
```

Creating a version of an object: Example

The following example can be used to understand how to create object versions.

1. Create a container with the **X-Versions-Location** header or add the header to an existing container.

In this example, `version_container` is the container that holds old versions of objects and `container1` is a new or existing container for which object versioning is to be enabled.

```
swift post -H "X-Versions-Location: version_container" container1
```

2. Run `swift stat` on `container1` to check that **X-Versions-Location** header is applied:

```
swift stat container1
Account: AUTH_f92886c4e3a347a18c29bae581b36788
Container: container1
Objects: 0
Bytes: 0
Read ACL:
Write ACL:
Sync To:
Sync Key:
Accept-Ranges: bytes
X-Storage-Policy: Policy-0
Connection: keep-alive
X-Timestamp: 1468226043.18746
X-Trans-Id: tx8d17476a914a40d781b0a-0057835a01
Content-Type: text/plain; charset=utf-8
X-Versions-Location: version_container
```

3. If *version_container* does not exist, create a new container:

```
swift post version_container
```

4. Run **swift list** at the account level to check that both containers are created successfully:

```
swift list
container1
version_container
```

5. Upload an object to *container1*:

```
swift upload container1 ImageA.jpg
```

6. Upload the second version of the object:

```
swift upload container1 ImageA.jpg
```

7. Upload the third version of the object:

```
swift upload container1 ImageA.jpg
```

8. Run **swift list** on the container to view the stored object:

```
swift list container1
ImageA.jpg
```

Note: The *container1* container contains only the most current version of the objects. The older versions of object are stored in *version_container*.

9. Run **swift list** on *version_container* to see the older versions of the object:

```
swift list version_container
00aImageA.jpg/1468227497.47123
00aImageA.jpg/1468227509.48065
```

10. To delete the most current version of the object, use the DELETE operation on the object:

```
swift delete container1 ImageA.jpg
ImageA.jpg
(deleted latest/third version)

# swift list container1
ImageA.jpg
(Second version is now the latest version)

# swift list version_container
00aImageA.jpg/1468227497.47123
(Initial version of the object)
```

Mapping of storage policies to filesets

For every storage policy created using the **mmobj policy create** command, one fileset is created or reused.

After a storage policy is created, you can specify that storage policy while creating new containers to associate that storage policy with those containers. When objects are uploaded into a container, they are stored in the fileset that is associated with the container's storage policy. For every new storage policy, a new object ring is created. The ring defines where objects are located and also defines multi-region replication settings.

The name of the fileset can be specified optionally as an argument of the **mmobj policy create** command. An existing fileset can be used only if it is not being used for an existing storage policy.

If even one of these prerequisites is missing, the **mmobj policy create** command fails. Otherwise, the fileset is used and the softlinks for the devices that are given to the ring builder point to it. If no fileset name is specified with the **mmobj policy create** command, a fileset is created using the policy name as a part of the fileset name with the prefix `obj_`.

For example, if a storage policy with name `Test` is created and no fileset is specified, a fileset with the name `obj_Test` is created and is linked to the base file system for object:

```
<object base filesystem mount point>/obj_Test/<n virt. Devices>
```



Attention: For any fileset that is created, its junction path is linked under the file system. The junction path should not be changed for a fileset that is used for a storage policy. If it is changed, data might be lost or it might get corrupted.

To enable swift to work with the fileset, softlinks under the given devices path in `object-server.conf` are created:

```
<devices path in object-server.conf>/<n virt. Devices>
<object base filesystem mount point>/obj_Test/<n virt. Devices>
```

Administering storage policies for Swift Object storage

Use the following information to create, list, and change storage policies for object storage.

Important:

- CES Swift Object protocol feature is not supported from IBM Storage Scale 5.2.0 onwards.
- IBM Storage Scale 5.1.8 is the last release that has CES Swift Object protocol.
- IBM Storage Scale 5.2.0 will tolerate the update of a CES node from IBM Storage Scale 5.1.8.
 - *Tolerate* means:
 - The CES node will be updated to 5.2.0.
 - Swift Object support will not be updated as part of the 5.2.0 update.
 - You may continue to use the version of Swift Object protocol that was provided in IBM Storage Scale 5.1.8 on the CES 5.2.0 node.
 - IBM will provide usage and known defect support for the version of Swift Object that was provided in IBM Storage Scale 5.1.8 until you migrate to a supported object solution that IBM Storage Scale provides.
 - CES Swift Object is replaced with IBM Storage Scale S3. For more details, refer the *S3 support overview* section the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
 - For more information about Swift Object in IBM Storage Scale, refer to the [IBM Storage Scale 5.2.0 documentation](#).
 - Please contact IBM for further details and migration planning.

Before you create a storage policy with the file-access (unified file and object access) function that is enabled, the file-access object capability must be enabled.

- To create a new storage policy with the unified file and object access feature that is enabled, run the following command:

```
mmobj policy create sof-policy --enable-file-access
```

A sample output is as follows:

```
[I] Getting latest configuration from ccr
[I] Creating fileset /dev/gpfso:obj_sof-policy
[I] Creating new unique index and build the object rings
[I] Updating the configuration
[I] Uploading the changed configuration
```

- To list storage policies for object storage with details of functions available with those storage policies, run the following command:

```
mmobj policy list --verbose
```

A sample output is as follows:

Index	Name	Deprecated Fileset	Fileset Path	Functions	Function Details	File System
0	SwiftDefault	object_fileset	/ibm/cesSharedRoot/object_fileset			cesSharedRoot
11751509160	sof-policy	obj_sof-policy	/ibm/cesSharedRoot/obj_sof-policy	file-and-object-access	regions="1"	cesSharedRoot
11751509230	mysofpolicy	obj_mysofpolicy	/ibm/cesSharedRoot/obj_mysofpolicy	file-and-object-access	regions="1"	cesSharedRoot

- To change a storage policy for object storage, run the following command:

```
mmobj policy change
```

- To change the default storage policy, run the following command:

```
mmobj policy change sof-policy --default
```

The system displays **sof-policy** as the default storage policy.

For more information about the **mmobj policy** command, see *mmobj command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Creating storage policy for object compression

Use the following information to create a storage policy with the compression function enabled and to create a storage policy with the compression schedule defined.

- To create a storage policy with the compression function enabled, use the **--enable-compression** option with the **mmobj policy create** command:

```
mmobj policy create CompressionTest --enable-compression --compression-schedule "MM:HH:dd:ww"
```

- To create a storage policy with the compression function enabled and a compression schedule that is defined, use the **--enable-compression** and the **--compression-schedule** options with the **mmobj policy create** command:

```
mmobj policy create CompressionTest --enable-compression --compression-schedule "MM:HH:dd:ww"
where
MM = 0-59 minutes
HH = 0-23 hours
dd = 1-31 day of month
ww = 0-7 (0=Sun, 7=Sun) day of week
```

- Use an asterisk (*) for specifying every instance of a unit. For example, dd = * means that the job is scheduled to run every day.

- Comma-separated lists are allowed. For example, dd = 1,3,5 means that the job is scheduled to run on every 1st, 3rd, 5th of a month.
- If ww and dd both are specified, the union is used.
- Specifying a range by using a dash (-) is not supported.
- Empty values are allowed for dd and ww. If empty, dd and or ww are not considered.
- Empty values for mm and hh are treated as *.

In the following example, the compression job is scheduled to run at 23:50 every day:

```
mmobj policy create CompressionTest --enable-compression --compression-schedule "50:23:*:*"
```

Every object that is stored by using a storage policy that is enabled is compressed according to the specified schedule. You do not need to decompress an object in advance of a get request or any other object request. IBM Storage Scale automatically returns the decompressed object.

Note:

- The download performance of objects in a compressed container is reduced compared to the download performance of objects in a non-compressed container.
- The compression function enables the file system compression over the object file set. The same compression functions and restrictions apply to object compression and file compression.

Related concepts

[File compression](#)

Creating storage policy for object encryption

Use the following information to create a storage policy for encryption.

- To create a storage policy with the encryption function enabled, use the **mmobj policy create** command:

```
mmobj policy create PolicyName -f FilesetName -i MaxNumInodes
--enable-encryption --encryption-keyfile EncryptionKeyName
--force-rule-append
```

where:

PolicyName

Indicates the name of the storage policy to create.

FilesetName

Indicates the fileset name that the created storage policy must use. This parameter is optional.

FilesystemName

Indicates the file system name where the fileset resides. This parameter is optional.

MaxNumInodes

Indicates the inode limit for the new inode space. This parameter is optional.

--enable-encryption

Enables an encryption policy.

EncryptionKeyName

Indicates the fully qualified path of the encryption key file.

--force-rule-append

Adds and establishes the rule when other rules exist. This parameter is optional.

The **--force-rule-append** determines whether to establish the GPFS policy rules:

- If **--force-rule-append** is not set:

1. The command checks whether a GPFS policy rule is already established during policy creation.
2. If the policy rule is established, the new encryption rule is not established but is displayed.

3. Otherwise, the new encryption rule is established and is displayed.
- If **--force-rule-append** is set:
 1. The command checks whether a GPFS policy rule is already established during policy creation.
 2. If the policy rule is established, the new encryption rule is added to the already established rules and the GPFS policy for the file system is updated. The new encryption rule is displayed.
 3. Otherwise, the new encryption rule is established and is displayed.

During command execution the encryption policy and rule are created. A GPFS policy rule file is created and used to establish the policy rule.

The following example shows a policy rule file: `/var/mmfs/ces/policyencryption.rule`

Note: The filename is autogenerated.

After the encryption policy is created, depending on the presence or absence of the **--force-rule-append** parameter, the command displays the new encryption policy.

If an error occurs during encryption, the local cleanup function is called to remove the created fileset and exit the CLI **mmobj** policy create script. The existing rules and policies are not changed.

Note: The encryption function enables the file system encryption over the object file set. The same encryption functions and restrictions apply to object encryption and file encryption.

Adding a region in a multi-region Swift Object deployment

Perform the following steps to add a region in a multi-region Swift Object deployment environment.

In the command examples, Europe is the first region and Asia is the second region.

1. Export the information of the first region to a file by using the **mmobj multiregion export** command:

```
[europe]# mmobj multiregion export --region-file /tmp/multiregion_europe.dat
```

2. Copy the file manually to the second region:

```
[europe]# scp /tmp/multiregion_europe.dat asia:/tmp
```

3. From the second region, join the multi-region environment as follows:

- a) Use the file that is generated in the first region while you deploy Swift Object in the second region by using the **mmobj swift base** command:

```
[asia]# mmobj swift base -g /mnt/gpfs0 --cluster-hostname gpfs-asia --pwd-file mmobjpwd
-i 100000 --admin-user admin \
--enable-multi-region --remote-keystone-url http://gpfs-asia:35357/v3
--join-region-file /tmp/multiregion_europe.dat \
--region-number 2 --configure-remote-keystone
```

This step installs the Swift Object protocol in the second region and it joins the first region. More devices are added to the primary ring files for this region.

4. Export the ring file data of the second region:

```
[asia]# mmobj multiregion export --region-file /tmp/multiregion_asia.dat
```

5. Copy the file manually to the first region:

```
[asia]# scp /tmp/multiregion_asia.dat europe:/tmp
```

6. In the first region, update the local ring files with the configuration of the second region:

```
[europe]# mmobj multiregion import --region-file /tmp/multiregion_asia.dat
```

This step reads in the ring files that are updated with the information of the second region. This update ensures that the data of the second region contains a new region. It also replaces the associated ring files in the first region with the ones from the second region.

Note:

The two clusters are now synced together and can be used as a multi-region cluster. Swift Objects can be uploaded and downloaded from either region. If the installation of the second region specified the **--configure-remote-keystone** flag, a region-specific endpoint for the Swift Object-store service for the second region is created in Keystone.

The regions must be synced in the future anytime region-related information changes. These changes include changes in the set of CES IP addresses (added or removed) or if storage policies were created or deleted within a region. Changes that affect the `swift.conf` file or ring files need to be synced to all regions. For example, adding more CES addresses to a region causes the ring files to be rebuilt.

7. In the second region, add CES addresses and update other clusters:

```
[asia]# mmces address add --ces-ip asia9
```

This step adds an address to the CES IP pool and triggers a ring rebuild that changes the IP-to-device mapping in the ring files.

8. Export the ring data so the other clusters in the region can be updated with the new IP addresses from the second region:

```
[asia]# mmobj multiregion export --region-file /tmp/multiregion_asia.dat
```

9. Copy the file manually to the first region:

```
[asia]# scp /tmp/multiregion_asia.dat europe:/tmp
```

10. In the first region, update with changes for the new second region address in the ring:

```
[europe]# mmobj multiregion import --region-file /tmp/multiregion_asia.dat
```

This step imports the changes from the second region. When the change import is complete, a checksum is displayed which can be used to determine when regions are synchronized together. By comparing it to the one printed when the region data was exported, you can determine that the regions are synchronized when they match. In some cases, the checksums do not match after import. The checksums do not match because some local configuration changes on this cluster that are not yet synced to the other regions. If the checksums do not match, then this region's configuration needs to be exported and imported into the other region to sync them.

Administering a multi-region object deployment environment

Use the following information to administer a multi-region object deployment environment.

A multi-region environment consists of several independent storage clusters that are linked together to provide unified object access. Configuration changes in one cluster that affect the multi-region environment are not automatically distributed to all clusters.

The cluster that makes the configuration change must export the relevant multi-region data. Then, the other regions must import that data to synchronize the multi-region configuration. Changes that affect multi-region are:

- Changes to the CES IP pool, such as adding or deleting addresses, which affect the ring layout.
- Changes to the object services ports used for the account, container, and object servers (ports 6200-6202).
- Creation, deletion, or modification of storage policies.
- Changes to the `swift.conf` configuration file.

Use the following commands to manage the configuration of the multi-region environment:

- To export the data for the current region so that it can be integrated into other regions, use the following command. The *RegionData* file that is created can be used to update other regions:

```
mmobj multiregion export --region-file RegionData
```

The *RegionData* file is created and it contains the updated multi-region information.

- To import the multi-region data to synchronize the configuration, use the following command. The *RegionData* must be the file that is created from the **mmobj multiregion export** command:

```
mmobj multiregion import --region-file RegionData
```

As part of the export or import commands, a region checksum is printed. This checksum can be used to ensure that the regions are synchronized. If the checksums values match, then the multi-region configuration of the clusters match. In some cases, the checksums do not match after import. The checksums do not match because the cluster that imports have local configuration changes that are synchronized with the other regions. For example, a storage policy was created but the multi-region configuration was not synchronized with the other regions. If that happens, the import command prints a message that the regions are not fully synchronized because of the local configuration and that the region data must be exported and imported to the other regions. After all regions have matching checksums, the multi-region environment is synchronized.

An existing region can be removed from the multi-region environment. This action permanently removes the region configuration, and the associated cluster cannot rejoin the multi-region environment.

The cluster of the removed region needs to disable object services because it is usable as a standalone object deployment.

- To remove a previously-defined region from the configuration, run the following command from a different region than the one being removed:

```
mmobj multiregion remove --remove-region-number RegionNumber
```

The cluster that is associated with the removed region must clean up object services with the **mmces service disable OBJ -a** command to uninstall object services.

- Run the following command to display the current multi-region information:

```
mmobj multiregion list
```

Unified file and Swift Object access in IBM Storage Scale

Unified file and Swift Object access allows use cases where you can access data by using Swift Object and file interfaces. Use the following information to manage unified file and Swift Object access that includes identity management modes for unified file and Swift Object access, authentication for unified file and Swift Object access, and objectization service schedule.

Important: In a unified file and Swift Object access environment, Swift Object ACLs apply only to accesses through the Swift Object interface and file ACLs apply only to accesses through the file interface.

For example, if user Bob ingests a file from the SMB interface and user Alice does not have access to that file from the SMB interface, it does not mean that Alice does not have access to the file from the Swift Object interface. The access rights of Alice for that file or Swift Object from the Swift Object interface depends on the ACL defined for Alice on the container in which that file or Swift Object resides.

Enabling object access to existing filesets

Learn to enable object access to files stored in an existing fileset.

[“How to enable access with --update-listing” on page 383](#)

[“How to enable access without --update-listing” on page 384](#)

[“How to enable access with a fileset path from a different object file system” on page 384](#)

Note: Before you enable object access for the existing filesets, ensure that SELinux is in the Permissive or Disabled mode.

How to enable access with --update-listing

This set of examples uses the following resources:

- The account name is admin.
- The policy name is sof_policy.
- The file system name is gpfs1.
- The fileset name is legacy_fset1.
- The fileset junction path is /gpfs1/legacy_fset1, which contains the following files:

```
existingfile1  
existingdir/existingfile2
```

- The container name is cont1.

You can do the following operations. All the commands are on one line:

1. The following command enables object access to the fileset and updates the container listing with the existing files:

```
mmobj file-access link-fileset --sourcefileset-path /gpfs1/legacy_fset1  
--account-name admin --container-name cont1 --fileaccess-policy-name sof_policy  
--update-listing
```

The command also creates the soft link gpfs1-legacy_fset1. The link name is constructed according to the following format: <file_system_name>-<fileset_name>.

2. Both of the following commands upload an object newobj to the linked fileset path /gpfs1/legacy_fset1. Both commands use the soft link gpfs1-legacy_fset1 that is created in the preceding example. You can use either method:

- The following example uses the swift utility:

```
swift upload -H "X-Storage-Policy: sof_policy" cont1 'gpfs1-legacy_fset1/newobj'
```

- The following example uses the curl utility:

```
curl -X PUT -T newobj -H "X-Storage-Policy: sof_policy"  
http://specscsaleswift.example.com:8080/v1/AUTH_cd1a29013b6842939a959dbda95835df/cont1/gpfs1-  
legacy_fset1/newobj
```

The following command creates a new directory newdir and uploads the object newobj1 to it:

```
swift upload -H "X-Storage-Policy: sof_policy" cont1 'gpfs1-legacy_fset1/newdir/newobj1'
```

3. The following command lists the contents of the container cont1:

```
swift list cont1
```

The command displays the following output:

```
gpfs1-legacy_fset1/newdir/newobj1  
gpfs1-legacy_fset1/newobj  
gpfs1-legacy_fset1/existingfile1  
gpfs1-legacy_fset1/existingdir/existingfile2
```

4. The following command downloads the file newobj:

```
swift download cont1 'gpfs1-legacy_fset1/newobj'
```

How to enable access without --update-listing

This set of examples uses the following resources:

- The account name is admin.
- The policy name is sof_policy.
- The file system name is gpfs1.
- The fileset name is legacy_fset2.
- The fileset junction path is /gpfs1/legacy_fset2, which contains the following file:
existingfile2
- The container name is cont2.

You can do the following operations. All the commands are on one line:

1. The following command enables object access to the fileset and creates the link gpfs1-legacy_fset2 but does not update the container listing with existing files:

```
mmobj file-access link-fileset --sourcefileset-path /gpfs1/legacy_fset2  
--account-name admin --container-name cont2 --fileaccess-policy-name sof_policy
```

2. The following command uploads the object newobj to the linked fileset path /gpfs1/legacy_fset2:

```
swift upload -H "X-Storage-Policy: sof_policy" cont2 'gpfs1-legacy_fset2/newobj'
```

3. The following command lists the contents of the container cont2:

```
swift list cont2
```

The command displays the following output.

```
gpfs1-legacy_fset2/newobj
```

The command displays only the objects that are added to the fileset, either by uploading the object or by specifying the --update-listing parameter with mmobj --file-access. Here the only such object is newobj. The command does not list the existing file existingfile2.

How to enable access with a fileset path from a different object file system

You can enable object access to an existing non-object fileset path where the fileset path is derived from a different object file system. To do so, omit the --update-listing parameter. You can access the data with the utilities swift or curl. However, the container listing is not updated with the existing file entries and object metadata is not appended to the existing data.

Identity management modes for unified file and Swift Object access

The following section gives information about the two identity management modes for unified file and Swift Object access: local mode and unified mode. The information in this section also describes how to configure these modes for a system.

Important:

- CES Swift Object protocol feature is not supported from IBM Storage Scale 5.2.0 onwards.
- IBM Storage Scale 5.1.8 is the last release that has CES Swift Object protocol.
- IBM Storage Scale 5.2.0 will tolerate the update of a CES node from IBM Storage Scale 5.1.8.
 - *Tolerate* means:
 - The CES node will be updated to 5.2.0.
 - Swift Object support will not be updated as part of the 5.2.0 update.

- You may continue to use the version of Swift Object protocol that was provided in IBM Storage Scale 5.1.8 on the CES 5.2.0 node.
- IBM will provide usage and known defect support for the version of Swift Object that was provided in IBM Storage Scale 5.1.8 until you migrate to a supported object solution that IBM Storage Scale provides.
- CES Swift Object is replaced with IBM Storage Scale S3. For more details, refer the *S3 support overview* section the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
- For more information about Swift Object in IBM Storage Scale, refer to the [IBM Storage Scale 5.2.0 documentation](#).
- Please contact IBM for further details and migration planning.

Unified file and Swift Object access comprises the following two modes:

- **local_mode:** Separate identity between Swift Object and file (Default mode)
- **unified_mode:** Shared identity between Swift Object and file

The mode is represented by the **id_mgmt** configuration parameter in the `object-server-sof.conf` file:

id_mgmt = local_mode | unified_mode

You can change this parameter by using the **mobj config change** command. For more information, see “[Configuring authentication and setting identity management modes for unified file and object access](#)” on page 398.

Note:

- Only one mode can be effective at a specified time and it must be configured by the administrator for the entire system. **id_mgmt = local_mode** is the default setting.
- If you plan to use **unified_mode**, the authentication mechanism for file and Swift Object must be the same. If you set **id_mgmt** to **unified_mode** and the file authentication and object authentication are not common, then the ID resolution of the users does not work correctly.

This leads to either Swift Object not being created with 503 error* return code or the Swift Object that is being created with an improper user ID. So, it is important that administrators make sure that a common authentication with appropriate ID mapping is configured for file and Swift Object.

* if you are using swift client, instead of 503, you might get an error similar to the following error:

```
'put_object('container_name', 'object_name', ...) failure and no ability to reset contents for reupload.'
```

For more information about validating the ID mapping, see “[Validating shared authentication ID mapping](#)” on page 390.

local_mode - separate identity between object and file

The following points must be considered when you plan to use **local_mode** identity management.

- Use-case for unified file and object access in **local_mode**:
 - Data that is created from the object interface is available for application to run analytics by using the file interface, where ownership of files is not essential.
 - Data that is created from the file interface is accessible from the object interface after objectization of those files.
- To address this use case, object authentication setup is independent of file authentication setup. Although, you can set up object and file authentication from a common authentication server for AD or LDAP.
- Objects that are created or updated by using the object interface are owned by the **swift** user. Applications processing the object data from file interface need the required file ACL to access the object data.

- Data updated from the file interface after objectization is available for object access.
- Containers that are created with a unified file and object access policy that are exposed as export points need appropriate ACLs set as needed by SMB, NFS, and POSIX.
- If the object exists, existing ownership of the corresponding file is retained if `retain_owner` is set to yes in `object-server-sof.conf`. For more information, see “[Configuration files for IBM Storage Scale for object storage](#)” on page 412.
- Retaining ACL, extended attributes (`xattrs`), and Windows attributes (`winattrs`): If the object is created or updated over existing file then existing file ACL, `xattrs`, and `winattrs` are retained if `retain_acl`, `retain_xattr`, and `retain_winattr` are set to yes in `object-server-sof.conf`. For more information, see “[Configuration files for IBM Storage Scale for object storage](#)” on page 412.

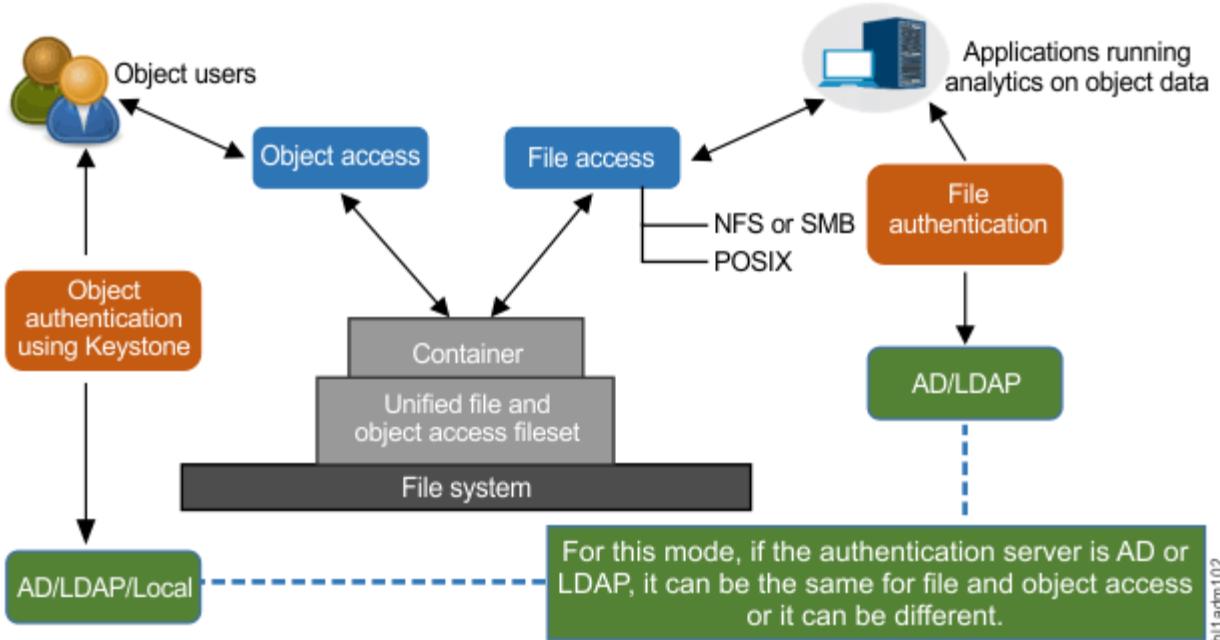


Figure 10. `local_mode` - separate identity between object and file

unified_mode - shared identity between object and file

Consider the following points when you use `unified_mode` identity management.

- Users from object and file are expected to be common and coming from the same directory service (only AD+RFC 2307 or LDAP).
- Note:** If your deployment uses only SMB-based file interface for file access and file authentication is configured with Active Directory (AD) with Automatic ID-mapping, unified file and object access can be used, assuming that object is configured with the same AD domain.
- Ownership: Object that is created from the object interface is owned by the user who completes the object PUT operation.
- If the object exists, existing ownership of the corresponding file is retained if `retain_owner` is set to yes in `object-server-sof.conf`. For more information, see “[Configuration files for IBM Storage Scale for object storage](#)” on page 412.
- Authorization: Object access follows the object ACL semantics and file access follows the file ACL semantics.
- Retaining ACL, extended attributes (`xattrs`), and Windows attributes (`winattrs`): If the object is created or updated over existing file then existing file ACL, `xattrs`, and `winattrs` are retained if `retain_acl`, `retain_xattr`, and `retain_winattr` are set to yes in `object-server-sof.conf`. For more information, see “[Configuration files for IBM Storage Scale for object storage](#)” on page 412.

- When a user does a PUT operation for an object over an existing object or does a PUT operation for a fresh object over a nested directory, no explicit file ACL is set for that user. If no explicit ACL is set for the user, it is possible that in some cases, the user might not have access to that file from the file interface even though the user has access from the object interface. This function prevents changing of the file ACL from the object interface to maintain file ACL semantics. In these cases, if the user is required to have permission to access the file also, explicit file ACL permission need to be set from the file interface.

For example, if user Bob completes a PUT operation for an object over an existing object (object maps to a file) owned by user Alice, Alice continues to own the file and there is no explicit file level ACL that is set for Bob for that file. Similarly, when Bob completes a PUT operation for a new object inside a subdirectory (already created by Alice), no explicit file ACL is set on the directory hierarchy for Bob. Bob does not have access to the object from the file interface unless there is an appropriate directory inheritance ACL that is set. To summarize, the object ingest does not change any file ACL and vice versa.

Table 24. Object input behavior in unified_mode.

Note: In the scenarios listed in the following table, the operations are being done by user Bob from the object interface. The instances that are owned by user Alice imply that the file or directory ownership maps to user Alice from the file side. Also, it is assumed that the `retain_owner`, `retain_acl`, `retain_xattr`, and `retain_winattr` parameters are set to yes in `object-server-sof.conf`.

Operation from SWIFT interface on object or container	Ownership result on corresponding file or directory		ACL, xattr, and winattr retention behavior on corresponding file or directory	
	File	Directory	File	Directory
Bob does a PUT operation for a new object.	The ownership of the file is set to Bob.	NA	Default GPFS ACLs are set	NA
Bob does a PUT operation for a new container.	NA	The ownership of the directory is set to Bob.	NA	Default GPFS ACLs are set
Bob does a PUT operation for an object that is already present and is owned by Alice.	The ownership of the file continues to be with Alice. Bob is not given any file ACL explicitly.	There are no changes in the ownership of the parent directory.	Existing ACL, file <code>xattrs</code> , and file <code>winattrs</code> are retained.**	NA
Bob does a POST operation (update metadata) of existing object that is owned by Alice.	The ownership of the file continues to be with Alice. Bob is not given any file ACL explicitly.	NA	Existing ACL, file <code>xattrs</code> , and file <code>winattrs</code> are retained.**	NA
Bob does a POST operation (update metadata or ACL) of existing container that is owned by Alice.	NA	The ownership of the directory continues to be with Alice. Bob is not given any directory ACL.	NA	NA
GET/DELETE/HEAD	No impact			

Note: **Unified file and object access retains the extended attributes (`xattr`), Windows attributes (`winattrs`) and ACL of the file if there is a PUT request from an object over an existing file. However, security or system namespace of extended attributes and other IBM Storage Scale extended attributes

such as immutability or pcache are not retained. Swift metadata (`user.swift.metadata`) is also not retained and it is replaced according to object semantics that is the expected behavior.

Advantages of using unified_mode

IBM Storage Scale offers various features that use user identity (UIDs or GIDs). With `unified_mode`, you can use these features seamlessly across file and object interfaces.

Unified access to object data

User can access object data by using SMB or NFS exports by using their AD or LDAP credentials.

Quota

GPFS quota for users that work on UID or GID can be set so they work for the file and object interface.

Example: User A can have X quota on a unified access fileset that is assigned by using GPFS quota commands that can hold true for all data that is created by the user from the file or the object interface.

For more information, see the [“Quota-related considerations for unified_mode” on page 389](#) section.

ILM

Tiering of user-specific data by using UID or GID.

Example 1

The UID and GID file attributes can be used to create an ILM placement policy to place the files that are owned by the Gold customers in faster storage pools and retain the files in the pools even when the pool storage starts reaching the threshold. The UID and GID file attributes can also be used to create a migration ILM policy so that, when the pool reaches its storage threshold, all files older than 30 days are moved to a slower storage pool except the ones owned by the Gold customers.

Example 2

After a user has left the organization, the UID of the user can be used to migrate the data and retain it on the archive tape when defined by the ILM policies.

Backup

Example: UID and GID file attributes in the policy rules that are defined for the `mmbbackup` command can be used to regularly back up the data of selective users.

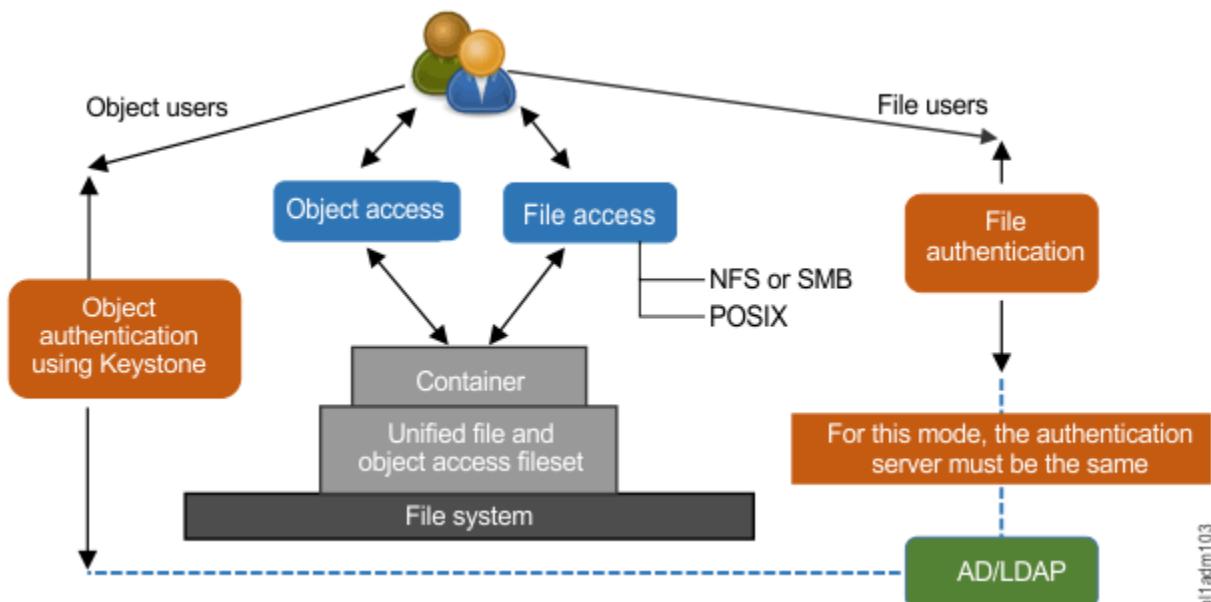


Figure 11. `unified_mode` - unified identity between object and file

bl1adm103

Quota-related considerations for unified_mode

There are three types of quotas that need to be considered:

- Consider the quota for a user set by using file system commands for that fileset that is set by using user ID or group ID. This quota represents the size in bytes up to which the user can create data on a specified fileset. This data is tracked at the file system level.
- Consider container quotas. The container quota is the size in bytes or number of objects that can be stored in a container regardless of the user that makes the upload (PUT) request. For more information, see [OpenStack documentation of container quotas](#).
- Consider account quotas. The account quota is the size in bytes or number of objects that can be stored in an account regardless of the user that makes the upload (PUT) request. For more information, see [OpenStack documentation of account quotas](#).

The fileset quotas, container level quotas, fileset quotas, and account quotas are independent of each other.

In some cases, the fileset quota must be cumulative of all the containers quotas that are hosted over it - though it is not mandatory. When both the quotas at the fileset level and at the container quota level are set (and if the fileset quota is reached), no more object data can be input on any of the containers that are hosted by that fileset - whether the container quota is reached. Hence, when you plan to use both file and object quotas, it is important to understand these details.

The objectization process does not consider the container quota and the account quota. This means that there might be a scenario where a container can host more data than the container quota associated with it especially when the **ibmobjectizer** service contains objectized files as objects.

For example, consider that:

- You want to have a total of 1 TB of data that is allocated for file and object access.
- You want each user to have an overall quota from the file and the object interface to be 10 GB.
- You have a pre-defined set of 100 containers that are enabled for object and file access (by using the storage policy for object storage) and users access to different containers depends on the container ACLs.

In this case, set the quotas:

1. Set the fileset quota that is associated with the file access policy to 1 TB.
2. Set the user quota on that fileset to 10 GB.
3. Set the container quota to the required level.

Remember: Setting it more than fileset quota cannot be accepted until fileset quota is increased or unset.

In this example scenario, the object access is restricted if either the user quota or the fileset quota is reached even though the container quota is not reached.

Authentication in unified file and object access

The following information provides information about how file authentication and object authentication are configured for different identity management modes.

Authentication configuration in local_mode: separate identity between object and file

In this mode, objects that are created continue to be owned by the swift user, which is an administrator under whose context the object server runs on the system. Object authentication can be configured to any supported authentication schemes since in this mode there is no ID mapping of objects to user ID. And, file authentication can continue to be configured to any supported authentication scheme.

For supported authentication schemes, see the *Authentication support matrix* table in the *Authentication considerations* topic in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Authentication configuration in unified_mode: shared identity between object and file

This mode allows objects and files to be owned by the users' UID and the corresponding GID that created them.

Important: Both the object protocol and the file protocol need to be configured with the same authentication scheme for this mode.

The supported authentication schemes for the unified mode are:

- AD for Authentication + RFC 2307 for ID mapping
- LDAP for authentication and for ID mapping

Note: User-defined authentication is not supported with both of the identity management modes.

Validating shared authentication ID mapping

Perform the following steps to validate shared authentication ID mapping.

1. List the authentication details on IBM Storage Scale by running the **mmuserauth service list** command.

The system displays output similar to the following output as follows:

```
FILE access configuration : LDAP
PARAMETERS VALUES
-----
ENABLE_SERVER_TLS false
ENABLE_KERBEROS false
USER_NAME cn=manager,dc=sonasldap,dc=com
SERVERS 9.118.37.234
NETBIOS_NAME deepakcluster
BASE_DN dc=sonasldap,dc=com
USER_DN dc=sonasldap,dc=com
GROUP_DN none
NETGROUP_DN none
USER_OBJECTCLASS posixAccount
GROUP_OBJECTCLASS posixGroup
USER_NAME_ATTRIB cn
USER_ID_ATTRIB uid
KERBEROS_SERVER none
KERBEROS_REALM none

OBJECT access configuration : LDAP
PARAMETERS VALUES
-----
ENABLE_ANONYMOUS_BIND false
ENABLE_SERVER_TLS false
ENABLE_KS_SSL false
USER_NAME cn=manager,dc=sonasldap,dc=com
SERVERS 9.118.37.234
BASE_DN dc=sonasldap,dc=com
USER_DN dc=sonasldap,dc=com
USER_OBJECTCLASS posixAccount
USER_NAME_ATTRIB cn
USER_ID_ATTRIB uid
USER_MAIL_ATTRIB mail
USER_FILTER none
ENABLE_KS_CASIGNING false
KS_ADMIN_USER userr
```

2. Make sure that the file authentication type and the object authentication type are the same. The valid values are AD and LDAP.

The following show potential file authentication and object authentication types:

```
FILE access configuration : LDAP
OBJECT access configuration : LDAP
```

With AD configuration, file authentication needs to be configured with Unix mapped domain. And the object authentication needs to also be configured with the same AD domain. This AD domain needs to be updated in the `object-server-sof.conf` configuration as:

```
ad_domain = <AD domain name>
```

3. Configure the file authentication and the object authentication against the same server as follows:

```
FILE : SERVERS 9.118.37.234  
OBJECT : SERVERS 9.118.37.234
```

Note: If there are multiple domain controllers in AD, the values might not match. The administrator needs to make sure that the server is referring to same user authentication source.

4. Make sure that the object users are receiving the correct UIDs and GIDs from the authentication source.

The following example uses `userr` as the object user:

```
cat /root/openrc  
export OS_AUTH_URL="http://127.0.0.1:35357/v3"  
export OS_IDENTITY_API_VERSION=3  
export OS_AUTH_VERSION=3  
export OS_USERNAME="userr"  
export OS_PASSWORD=""  
export OS_USER_DOMAIN_NAME=Default  
export OS_PROJECT_NAME=admin  
export OS_PROJECT_DOMAIN_NAME=Default
```

5. Make sure that the object user is correctly resolved on all the protocol nodes and the same UID and GID are listed.

The following example lists the UID and GID for the object user `userr`:

```
id userr  
uid=1101(userr) gid=1000(testgrp) groups=1000(testgrp),1002(testgrp2)
```

The objectizer process

The objectization process converts files ingested from the file interface on unified file and access enabled container path to be available from the object interface. The name of that service is **ibmobjectizer**.

When new files are added from the file interface, they need to be visible to the Swift database to show correct container listing and container or account statistics.

The **ibmobjectizer** service provides synchronization between the file metadata and the object metadata at a predefined time interval (that assists with accurate container and account listing). The **ibmobjectizer** service identifies new files added from the file interface and adds the Swift system metadata to them so that they are objectized. The **ibmobjectizer** service then determines its container and account databases and adds an object entry to them. It also identifies files deleted from file interface and deletes their corresponding entries from container and account databases.

These functions are useful in the setups where data is ingested using legacy file interface-based devices such as medical and scientific devices. They are helpful when data needs to be stored and accessed over cloud using the object interface.

The **ibmobjectizer** service is a singleton and it is started when object is enabled and the file-access object capability is set. However, the **ibmobjectizer** service starts objectization only when there are containers with unified file and object access storage policies configured and the file-access object capability is set.

There are use cases in which objects are ingested using the object interface and the file interface is used only for reading them. For these use cases, the **ibmobjectizer** service is not needed. It can be disabled using the `mmobj file-access` command. For more information, see [“Starting and stopping the ibmobjectizer service” on page 395](#).

Run the following command to identify the node on which the **ibmobjectizer** service is running:

```
mmces service list --verbose
```

Run the following command when you have a cluster that has gpfssnode3 as the object singleton node:

```
mmces service list --verbose -a | grep ibmobjectizer
```

The system displays the following output:

```
gpfssnode3:          OBJ:ibmobjectizer           is running
```



Attention: If object services on the singleton node are stopped by the administrator manually, objectization is stopped across the cluster. Manually stopping services on a singleton node need to be planned carefully after understanding the impact.

For information on limitations on the objectizer process, see [“Limitations of unified file and object access” on page 408](#).

Related concepts

[“Understanding and managing Object services” on page 367](#)

Use the following information to manage services that are related to IBM Storage Scale for Object storage.

Related tasks

[“Setting up the objectizer service interval” on page 396](#)

Take the following steps to set up the objectizer service interval.

Related reference

[“Configuration files for IBM Storage Scale for object storage” on page 412](#)

Use the following information to manage options in configuration files that are used for IBM Storage Scale for object storage that includes the unified file and object access feature. These configuration files are located in the /etc/swift directory.

File path in unified file and Swift Object access

One of the key advantages of unified file and Swift Object access is the placement and naming of Swift Objects when they are stored on the file system.

Important:

- CES Swift Object protocol feature is not supported from IBM Storage Scale 5.2.0 onwards.
- IBM Storage Scale 5.1.8 is the last release that has CES Swift Object protocol.
- IBM Storage Scale 5.2.0 will tolerate the update of a CES node from IBM Storage Scale 5.1.8.
 - *Tolerate* means:
 - The CES node will be updated to 5.2.0.
 - Swift Object support will not be updated as part of the 5.2.0 update.
 - You may continue to use the version of Swift Object protocol that was provided in IBM Storage Scale 5.1.8 on the CES 5.2.0 node.
 - IBM will provide usage and known defect support for the version of Swift Object that was provided in IBM Storage Scale 5.1.8 until you migrate to a supported object solution that IBM Storage Scale provides.
 - CES Swift Object is replaced with IBM Storage Scale S3. For more details, refer the *S3 support overview* section the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
 - For more information about Swift Object in IBM Storage Scale, refer to the [IBM Storage Scale 5.2.0 documentation](#).
 - Please contact IBM for further details and migration planning.

Unified file and Swift Object access stores Swift Objects following the same path hierarchy as the Swift Object's URL. In contrast, the default Swift Object implementation stores the Swift Object following the mapping given by the ring, and its final file path cannot be determined by the user easily.

A Swift Object with the following URL is stored by the two systems as follows:

- **Swift Object URL:** `https://swift.example.com/v1/acct/cont/obj`
- **Path in default Swift Object implementation:** `/ibm/gpfs0/object_fileset/o/z1device108/objects/7551/125/75fc66179f12dc513580a239e92c3125/75fc66179f12dc513580a239e92c3125.data`
- **Path in unified file and Swift Object access:** `/ibm/gpfs0/obj_sofpolicy1/s69931509221z1device1/AUTH_763476384728498323747/cont/obj`

It is assumed that the Swift Object is configured over the `/ibm/gpfs0` file system with the default Swift Object on the `object_fileset` fileset and the unified file and Swift Object access data is located under the `obj_sofpolicy1` fileset. `s69931509221z1device1` is auto-generated based on the swift ring parameters and `AUTH_763476384728498323747` is auto-generated based on the account ID from keystone.



Attention: Do not unlink Swift Object filesets - including the unified file and Swift Object access enabled filesets.

Determining the POSIX path of a unified file and object access enabled fileset

Use the following steps for determining the POSIX path of a unified file and object access enabled fileset.

1. List all storage policies for object.

mmobj policy list						
Index	Name	Default	Deprecated	Fileset	Functions	File System
0	SwiftDefault	yes		object_fileset		
13031510160	sof-policy1			obj_sof-policy1	file-and-object-access	fs0
13031510260	CompressionTest		yes	obj_CompressionTest	compression	fs0
13031510290	CompressionDebug		yes	obj_CompressionDebug	compression	fs0
13031511020	CompressionNew			obj_CompressionNew	compression	fs0

2. In the `fs0` file system, note the index and fileset name for the policy you want. Run the `mmlsfileset` command to determine the junction point.

```
mmlsfileset fs0 | grep obj_sof-policy1  
obj_sof-policy1      Linked      /ibm/fs0/obj_sof-policy1
```

The Swift ring builder creates a single virtual device for unified file and object access policies. This virtual device is named with storage policy index number, which is also the region number. It starts with `s` and appended with `z1device1`.

```
s13031510160z1device1
```

3. List the Swift projects and identify the one you are interested in working with:

```
~/openrc  
openstack project list  
+-----+-----+  
| ID   | Name  |  
+-----+-----+  
| 73282e8bca894819a3cf19017848ce6b | admin |  
| 1f78f58572f746c39247a27c1e0e1488 | service |  
+-----+-----+
```

4. Construct the account name by appending the project ID with `AUTH_`. Or, substitute the correct project prefix when you have customized the prefix. For the admin project, use:

```
AUTH_73282e8bca894819a3cf19017848ce6b
```

The full path to the unified file and object access containers is the concatenation of the fileset linkage, the virtual device name, and the account name:

```
/fileset linked path/s<policy_number>z1device1/AUTH_account id/
```

A possible file path might be as follows:

```
/ibm/fs0/obj_sof-policy1/s13031510160z1device1/AUTH_73282e8bca894819a3cf19017848ce6b/
```

5. List the containers defined for this account.

```
ls /ibm/fs0/obj_sof-policy1/s13031510160z1device1/AUTH_73282e8bca894819a3cf19017848ce6b/  
new1      fifthcontainer  RTC73189_1  RTC73189_3  RTC73189_5  RTC73189_7  sixthcontainer
```

Administering unified file and object access

Use the following information to administer unified file and object access in your IBM Storage Scale setup.

Important:

- CES Swift Object protocol feature is not supported from IBM Storage Scale 5.2.0 onwards.
- IBM Storage Scale 5.1.8 is the last release that has CES Swift Object protocol.
- IBM Storage Scale 5.2.0 will tolerate the update of a CES node from IBM Storage Scale 5.1.8.
 - *Tolerate* means:
 - The CES node will be updated to 5.2.0.
 - Swift Object support will not be updated as part of the 5.2.0 update.
 - You may continue to use the version of Swift Object protocol that was provided in IBM Storage Scale 5.1.8 on the CES 5.2.0 node.
 - IBM will provide usage and known defect support for the version of Swift Object that was provided in IBM Storage Scale 5.1.8 until you migrate to a supported object solution that IBM Storage Scale provides.
 - CES Swift Object is replaced with IBM Storage Scale S3. For more details, refer to the *S3 support overview* section the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
 - For more information about Swift Object in IBM Storage Scale, refer to the [IBM Storage Scale 5.2.0 documentation](#).
- Please contact IBM for further details and migration planning.

Enabling the file-access object capability

Before you can use unified file and object access, you must enable the file-access object capability on the whole cluster.

- To enable the file-access object capability, enter the following command:

```
mmobj file-access enable
```

- To verify that the file-access object capability is enabled, enter the `mmobj config list` command, as in the following example:

```
mmobj config list --ccrfile spectrum-scale-object.conf --section capabilities --property file-access-enabled
```

The system displays output similar to the following:

```
file-access-enabled = true
```

Related tasks

[Starting and stopping the ibmobjectizer service](#)

The following information provides the commands to start and stop the ibmobjectizer service.

Setting up the objectizer service interval

Take the following steps to set up the objectizer service interval.

Enabling and disabling QOS

This topic lists the commands to enable and disable QOS.

Configuring authentication and setting identity management modes for unified file and object access

You can configure authentication and set the identity management modes for unified file and object access using the following steps.

Creating or using a unified file and object access storage policy

Use the following steps to create or use a unified file and object access storage policy.

Associating containers with a unified file and object access storage policy

Use the following steps to associate a container with a unified file and object access storage policy.

Creating exports on a container that is associated with a unified file and object access storage policy

Use the following steps to create a Network File System (NFS) or Server Message Block (SMB) export on the directory that maps to the container associated with the unified file and object access storage policy.

Enabling object access for selected files

Use the following steps to objectize files under all containers that are associated with the unified file and object access storage policy under a specified account.

Example scenario - administering unified file and object access

The following example describes an end-to-end scenario of administering and configuring unified file and object access.

Starting and stopping the ibmobjectizer service

The following information provides the commands to start and stop the ibmobjectizer service.

Run the following commands to start and stop the ibmobjectizer service:

Note: The ibmobjectizer service starts when **file-access-enabled** is set to true.

- Run the following command to start the ibmobjectizer service when it has stopped:

```
mmobj file-access enable
```

- Run the following command to stop the ibmobjectizer service:

```
mmobj file-access disable --objectizer
```

- Run the following command to check the service status of the ibmobjectizer service:

```
mmces service list -v -a | grep ibmobjectizer
```

Related tasks

Enabling the file-access object capability

Before you can use unified file and object access, you must enable the file-access object capability on the whole cluster.

Setting up the objectizer service interval

Take the following steps to set up the objectizer service interval.

Enabling and disabling QOS

This topic lists the commands to enable and disable QOS.

Configuring authentication and setting identity management modes for unified file and object access

You can configure authentication and set the identity management modes for unified file and object access using the following steps.

Creating or using a unified file and object access storage policy

Use the following steps to create or use a unified file and object access storage policy.

[Associating containers with a unified file and object access storage policy](#)

Use the following steps to associate a container with a unified file and object access storage policy.

[Creating exports on a container that is associated with a unified file and object access storage policy](#)

Use the following steps to create a Network File System (NFS) or Server Message Block (SMB) export on the directory that maps to the container associated with the unified file and object access storage policy.

[Enabling object access for selected files](#)

Use the following steps to objectize files under all containers that are associated with the unified file and object access storage policy under a specified account.

[Example scenario - administering unified file and object access](#)

The following example describes an end-to-end scenario of administering and configuring unified file and object access.

Setting up the objectizer service interval

Take the following steps to set up the objectizer service interval.

The default interval between the completion of an objectizer cycle and the starting of the next cycle is 30 minutes. However, it needs to be planned properly based on the following information:

- The frequency and the number of new file ingestions that might be objectized.
- The number of protocol nodes that are deployed.
- How quickly the ingested file needs to be objectized.

Remember: Objectization is a resource-intensive process.

The resource utilization is related to the number of containers that have unified file and object access enabled. The schedule of running the objectization process needs to be planned carefully. Running it too frequently might impact your protocol node's resource utilization. Schedule it during off business hours, especially when there are only a few protocol nodes (such as 2) with basic resource configuration. Or, schedule it with an interval of 30 minutes or more when you have protocol nodes with adequate resources (where the number of protocol nodes is more than 2).

Remember: Set the objectizer service interval to a no less than 30 minutes irrespective of the setup. If you need to urgently objectize files, you can use the **mmobj file-access** command to objectize the specified files.

- Set up the objectization interval by using the **mmobj config change** as follows:

```
mmobj config change --ccrfile spectrum-scale-objectizer.conf \
--section DEFAULT --property objectization_interval --value 2400
```

This command sets an interval of 40 minutes between the completion of an objectization cycle and the start of the next cycle.

- Verify that the objectization time interval is changed using the **mmobj config list** as follows:

```
mmobj config list --ccrfile spectrum-scale-objectizer.conf --section DEFAULT
--property objectization_interval
```

Related tasks

[Enabling the file-access object capability](#)

Before you can use unified file and object access, you must enable the file-access object capability on the whole cluster.

[Starting and stopping the ibmobjectizer service](#)

The following information provides the commands to start and stop the ibmobjectizer service.

[Enabling and disabling QOS](#)

This topic lists the commands to enable and disable QOS.

[Configuring authentication and setting identity management modes for unified file and object access](#)

You can configure authentication and set the identity management modes for unified file and object access using the following steps.

Creating or using a unified file and object access storage policy

Use the following steps to create or use a unified file and object access storage policy.

Associating containers with a unified file and object access storage policy

Use the following steps to associate a container with a unified file and object access storage policy.

Creating exports on a container that is associated with a unified file and object access storage policy

Use the following steps to create a Network File System (NFS) or Server Message Block (SMB) export on the directory that maps to the container associated with the unified file and object access storage policy.

Enabling object access for selected files

Use the following steps to objectize files under all containers that are associated with the unified file and object access storage policy under a specified account.

Example scenario - administering unified file and object access

The following example describes an end-to-end scenario of administering and configuring unified file and object access.

Enabling and disabling QOS

This topic lists the commands to enable and disable QOS.

The periodic scans run by the ibmobjectizer service are resource intensive and might affect the object IO performance. Quality Of Service (QOS) can be set on the ibmobjectizer service depending upon the IO workload and the priority at which the ibmobjectizer service must be run. The usage of resources is limited to the given number so that other high priority workflows and processes can continue with adequate resources, thereby maintaining the performance of the system.

- To enable QOS, type `mmchqos <fs> --enable`.
- Set the **qos_iops_target** parameter in the spectrum-scale-objectizer.conf file.

The following example is on one line:

```
mmobj config change --ccrfile spectrum-scale-objectizer.conf --section DEFAULT --property qos_iops_target --value 400
```

- To disable QOS on ibmobjectizer, set the **qos_iops_target** to 0.

The following example is on one line:

```
mmobj config change --ccrfile spectrum-scale-objectizer.conf --section DEFAULT --property qos_iops_target --value 0
```

Related tasks

Enabling the file-access object capability

Before you can use unified file and object access, you must enable the file-access object capability on the whole cluster.

Starting and stopping the ibmobjectizer service

The following information provides the commands to start and stop the ibmobjectizer service.

Setting up the objectizer service interval

Take the following steps to set up the objectizer service interval.

Configuring authentication and setting identity management modes for unified file and object access

You can configure authentication and set the identity management modes for unified file and object access using the following steps.

Creating or using a unified file and object access storage policy

Use the following steps to create or use a unified file and object access storage policy.

Associating containers with a unified file and object access storage policy

Use the following steps to associate a container with a unified file and object access storage policy.

Creating exports on a container that is associated with a unified file and object access storage policy

Use the following steps to create a Network File System (NFS) or Server Message Block (SMB) export on the directory that maps to the container associated with the unified file and object access storage policy.

Enabling object access for selected files

Use the following steps to objectize files under all containers that are associated with the unified file and object access storage policy under a specified account.

Example scenario - administering unified file and object access

The following example describes an end-to-end scenario of administering and configuring unified file and object access.

Configuring authentication and setting identity management modes for unified file and object access

You can configure authentication and set the identity management modes for unified file and object access using the following steps.

The identity management modes for unified file and object access are set in the `object-server-sof.conf` file. The default mode is `local_mode`.

Note: It is important to understand the identity management modes for unified file and object access and set the mode you want accordingly. Although it is possible to move from one mode to another, some considerations apply in that scenario.

The `unified_mode` identity management mode for unified file and object access is supported only with Active Directory (AD) with UNIX-mapped domains and LDAP authentication configurations. This mode must not be configured with local or user-defined authentication configurations.

Important: If you are using `unified_mode`, the authentication for both file and object access must be configured and the authentication schemes must be the same and configured with the same server. If not, the request to create object might fail with user not found error.

Use the following steps on a protocol node to configure authentication and enable `unified_mode`.

1. Determine which authentication scheme best suits your requirements. You can use either LDAP or AD with UNIX-mapped domains.

Note: Because object can be configured with only one AD domain, you need to plan which of the UNIX-mapped AD domains, in case there are trusted domains, is to be configured for object.

2. Configure file access using the `mmuserauth` command as follows.

```
mmuserauth service create --data-access-method file  
--type ad --servers myADserver --idmap-role master  
--netbios-name scale --unixmap-domains 'DOMAIN(5000-20000)'
```

3. Configure object access using the `mmuserauth` command as follows.

```
mmuserauth service create --data-access-method object --type ad  
--user-name "cn=Administrator,cn=Users,dc=IBM,dc=local"  
--base-dn "dc=IBM,DC=local" --ks-dns-name c40bbc2xn3 --ks-admin-user admin  
--servers myADserver --user-id-attrib cn --user-name-attrib sAMAccountName  
--user-objectclass organizationalPerson --user-dn "cn=Users,dc=IBM,dc=local"  
--ks-swift-user swift
```

4. Change `id_mgmt` in the `object-server-sof.conf` file using the `mmobj config change` command as follows.

```
mmobj config change --ccrfile object-server-sof.conf --section DEFAULT  
--property id_mgmt --value unified_mode
```

5. If object authentication is configured with AD, set `ad_domain` in the `object-server-sof.conf` file.

```
mmobj config change --ccrfile object-server-sof.conf --section DEFAULT  
--property ad_domain --value POLLUX
```

Note: Do not specify `ad_domain` with LDAP configurations.

To find the correct **ad_domain** name, use the following command:

```
/usr/lpp/mmfs/bin/net ads lookup -S {AD_SERVER_NAME | AD_SERVER_IP} -d0
```

For example, in the output of the following command, the value of the **Pre-Win2k Domain** field is the **ad_domain**.

```
/usr/lpp/mmfs/bin/net ads lookup -S 192.196.79.34 -d0

...
Forest:          pollux.com
Domain:         pollux.com
Domain Controller:  win2k8.pollux.com
Pre-Win2k Domain: POLLUX
Pre-Win2k Hostname:  WIN2K8
Server Site Name : Default-First-Site-Name
Client Site Name : Default-First-Site-Name
...
```

Your unified file and object access enabled fileset is now configured with **unified_mode**.

6. List the currently configured **id_mgmt** mode using the **mmobj config list** command as follows.

```
mmobj config list --ccrfile object-server-sof.conf --section DEFAULT --property id_mgmt
```

Important:

1. If the PUT requests fail in **unified_mode**, check if the user name is resolvable on the protocol nodes using the following command:

```
id '<user_name>'
```

If user name in AD is in the domain\user_name format, use the following command:

```
id '<domain\><user_name>'
```

2. Ensure that the **ad_domain** parameter is not present in the **object-server-sof.conf** file when LDAP is configured.

- To list the **object-server-sof.conf** file contents, use the following command:

```
mmobj config list --ccrfile object-server-sof.conf
```

- If **ad_domain** is present, remove it as follows:

- a. Copy **/etc/swift/object-server-sof.conf** to a temporary location, say **/tmp**.
- b. Modify the temporary file by appending a '**-**' before the **ad_domain** parameter. This marks that parameter for deletion.
- c. Upload the modified file using the following command:

```
mmobj config change --ccrfile object-server-sof.conf --merge-file /tmp/object-server-sof.conf
```

- d. **[Optional]:** Validate that **ad_domain** is removed from the **object-server-sof.conf** file by listing the file contents.

3. Configuring file authentication with the same scheme as that of object authentication is a mandatory prerequisite before you enable the **unified_mode** identity management mode. In case you configure file authentication later, you must restart swift on the file server for the changes to be effective. You can do this by changing **id_mgmt** to **local_mode** and then changing it back to **unified_mode** using the following commands.

```
mmobj config change --ccrfile object-server-sof.conf --section DEFAULT
--property id_mgmt --value local_mode
mmobj config change --ccrfile object-server-sof.conf --section DEFAULT
--property id_mgmt --value unified_mode
```

Related tasks

[Enabling the file-access object capability](#)

Before you can use unified file and object access, you must enable the file-access object capability on the whole cluster.

[Starting and stopping the ibmobjectizer service](#)

The following information provides the commands to start and stop the ibmobjectizer service.

[Setting up the objectizer service interval](#)

Take the following steps to set up the objectizer service interval.

[Enabling and disabling QOS](#)

This topic lists the commands to enable and disable QOS.

[Creating or using a unified file and object access storage policy](#)

Use the following steps to create or use a unified file and object access storage policy.

[Associating containers with a unified file and object access storage policy](#)

Use the following steps to associate a container with a unified file and object access storage policy.

[Creating exports on a container that is associated with a unified file and object access storage policy](#)

Use the following steps to create a Network File System (NFS) or Server Message Block (SMB) export on the directory that maps to the container associated with the unified file and object access storage policy.

[Enabling object access for selected files](#)

Use the following steps to objectize files under all containers that are associated with the unified file and object access storage policy under a specified account.

[Example scenario - administering unified file and object access](#)

The following example describes an end-to-end scenario of administering and configuring unified file and object access.

Creating or using a unified file and object access storage policy

Use the following steps to create or use a unified file and object access storage policy.

1. Run the following command to create a unified file and object access storage policy (and create a filesset):

```
mmobj policy create sof-policy1 --enable-file-access
```

The system output displays as follows:

```
[I] Getting latest configuration from ccr
[I] Creating filesset gpfso:obj_sof-policy1
[I] Creating new unique index and building the object rings
[I] Updating the configuration
[I] Uploading the changed configuration
```

2. List the available storage policies using the **mmobj policy list** command and determine which policies are for unified file and object access by viewing the **Functions** column of the output:

```
mmobj policy list --verbose
```

The system output displays as follows:

Index	Name	Deprecated	Fileset	Fileset Path	Functions	Function Details
0	SwiftDefault		object_fileset	/ibm/cesSharedRoot/object_fileset		
11751509160	sof-policy1		obj_sof-policy1	/ibm/cesSharedRoot/obj_sof-policy1	file-and-object-access	regions="1"
11751509230	mysofpolcy		obj_mysofpolcy	/ibm/cesSharedRoot/obj_mysofpolcy	file-and-object-access	regions="1"
11751510260	Test19		obj_Test19	/ibm/cesSharedRoot/obj_Test19		regions="1"

3. Use one of these storage policies to create data in a unified file and object access environment.

For more information, see “[Associating containers with a unified file and object access storage policy](#)” on page 401 and “[Creating exports on a container that is associated with a unified file and object access storage policy](#)” on page 402.

You can learn more about mapping storage policy and filesets. For more information, see “[Mapping of storage policies to filesets](#)” on page 377.

You must create export at the container level. From NFS or SMB, if you create a peer container, base containers that are created from NFS and SMB cannot be multiprotocol.

Related tasks

[Enabling the file-access object capability](#)

Before you can use unified file and object access, you must enable the file-access object capability on the whole cluster.

[Starting and stopping the ibmobjectizer service](#)

The following information provides the commands to start and stop the ibmobjectizer service.

[Setting up the objectizer service interval](#)

Take the following steps to set up the objectizer service interval.

[Enabling and disabling QOS](#)

This topic lists the commands to enable and disable QOS.

[Configuring authentication and setting identity management modes for unified file and object access](#)

You can configure authentication and set the identity management modes for unified file and object access using the following steps.

[Associating containers with a unified file and object access storage policy](#)

Use the following steps to associate a container with a unified file and object access storage policy.

[Creating exports on a container that is associated with a unified file and object access storage policy](#)

Use the following steps to create a Network File System (NFS) or Server Message Block (SMB) export on the directory that maps to the container associated with the unified file and object access storage policy.

[Enabling object access for selected files](#)

Use the following steps to objectize files under all containers that are associated with the unified file and object access storage policy under a specified account.

[Example scenario - administering unified file and object access](#)

The following example describes an end-to-end scenario of administering and configuring unified file and object access.

Associating containers with a unified file and object access storage policy

Use the following steps to associate a container with a unified file and object access storage policy.

1. Run the following command to export common environment variables by sourcing the openrc file:

```
source ~/openrc
```

2. Run the following command to associate a container with a unified file and object access storage policy:

```
swift post container1 --header "X-Storage-Policy: sof-policy1"
```

In this **swift post** example, the storage policy is specified with the customized header X-Storage-Policy using the --header option.

3. Run the following command to upload an object in the container that is associated with the unified file and object access storage policy:

```
swift upload container1 imageA.JPG
```

Note: The steps that are done by using **swift** commands can also be done by using **curl** commands. For more information, see “[curl commands for unified file and object access related user tasks](#)” on page 411.

Related tasks

[Enabling the file-access object capability](#)

Before you can use unified file and object access, you must enable the file-access object capability on the whole cluster.

Starting and stopping the ibmobjectizer service

The following information provides the commands to start and stop the ibmobjectizer service.

Setting up the objectizer service interval

Take the following steps to set up the objectizer service interval.

Enabling and disabling QOS

This topic lists the commands to enable and disable QOS.

Configuring authentication and setting identity management modes for unified file and object access

You can configure authentication and set the identity management modes for unified file and object access using the following steps.

Creating or using a unified file and object access storage policy

Use the following steps to create or use a unified file and object access storage policy.

Creating exports on a container that is associated with a unified file and object access storage policy

Use the following steps to create a Network File System (NFS) or Server Message Block (SMB) export on the directory that maps to the container associated with the unified file and object access storage policy.

Enabling object access for selected files

Use the following steps to objectize files under all containers that are associated with the unified file and object access storage policy under a specified account.

Example scenario - administering unified file and object access

The following example describes an end-to-end scenario of administering and configuring unified file and object access.

Creating exports on a container that is associated with a unified file and object access storage policy

Use the following steps to create a Network File System (NFS) or Server Message Block (SMB) export on the directory that maps to the container associated with the unified file and object access storage policy.

Create an SMB or NFS export on the directory that maps to the container associated with the unified file and object access storage policy.

- a) Run the following command to create the NFS export:

```
mmnfs export add "/ibm/gpfs0/obj_sofpolicy1/s69931509221z1device1/AUTH_763476384728498323747/  
cont"
```

- b) Run the following command to create the SMB share:

```
mmsmb export add smbexport "/ibm/gpfs0/obj_sofpolicy1/s69931509221z1device1/AUTH_763476384728498323747/  
cont"
```

Note:

- It is recommended that you create file exports on or below the container path level and not above it.

Important: Creating file exports above the container path level might lead to deletion of the unified file and object access enabled containers that is undesirable.

- When you use the POSIX interface, it is recommended to allow access only of data to POSIX users from on or below the container path.

Important: Accidental deletion of container or data above might lead to inconsistent state of the system.

Related tasks

Enabling the file-access object capability

Before you can use unified file and object access, you must enable the file-access object capability on the whole cluster.

Starting and stopping the *ibmobjectizer* service

The following information provides the commands to start and stop the *ibmobjectizer* service.

Setting up the objectizer service interval

Take the following steps to set up the objectizer service interval.

Enabling and disabling QOS

This topic lists the commands to enable and disable QOS.

Configuring authentication and setting identity management modes for unified file and object access

You can configure authentication and set the identity management modes for unified file and object access using the following steps.

Creating or using a unified file and object access storage policy

Use the following steps to create or use a unified file and object access storage policy.

Associating containers with a unified file and object access storage policy

Use the following steps to associate a container with a unified file and object access storage policy.

Enabling object access for selected files

Use the following steps to objectize files under all containers that are associated with the unified file and object access storage policy under a specified account.

Example scenario - administering unified file and object access

The following example describes an end-to-end scenario of administering and configuring unified file and object access.

Enabling object access for selected files

Use the following steps to objectize files under all containers that are associated with the unified file and object access storage policy under a specified account.

In a unified file and object access environment, you can access files that are created from file interfaces such as POSIX, NFS, or CIFS through object interfaces such as curl or Swift. But, you need to make these files available for the object interface.

To make these files available for the object interface after it is activated, the *ibmobjectizer* service runs periodically and makes newly created files available for the object interface. You can also use the **mmobj file-access** command to selectively enable files for access through the object interface immediately without waiting for the objectization time interval.

The purpose of this command is to make certain files available to object sooner (or immediately) than when the objectizer makes them available. This command does not ensure synchronization between file and object data. Therefore, files that are deleted are not immediately reflected in the object interface. Complete synchronization is done by the *ibmobjectizer* service eventually.

In unified file and object access enabled filesets, you can access files from the object interface if you know the entire URI (including keystone account ID, device, and other details). You can then access that file without the need for them to be objectized either by using the *ibmobjectizer* service or this **mmobj file-access** command (such as in the following examples).

Note: Disabling object access for files is not supported.

- Run the following command to objectize files under all the containers that are associated with the unified file and object access storage policy under an account:

```
mmobj file-access objectize --storage-policy sof_policy --account-name admin
```

The system output displays as follows:

```
Loading objectization configuration from CCR
Fetching storage policy details
Performing objectization
Objectization complete
```

This command objectizes all containers in the account admin and enables them for access through the object interface.

- Run the following command to objectize files under a container:

```
mmobj file-access objectize --storage-policy sof_policy --account-name admin --container-name container1
```

This command objectizes all files in container1 and enables them for access through the object interface.

- Run the following command to objectize a file while you specify a storage policy:

```
mmobj file-access objectize --storage-policy sof_policy --account-name admin \
--container-name container1 --object-name file1.txt
```

This command objectizes file1.txt in container1 and enables it for access through the object interface.

- Run the following command to objectize a file:

```
mmobj file-access objectize --object-path \
/ibm/gpfs0/obj_sofpolicy1/s69931509221z1device1/AUTH_763476384728498323747/cont/file1.txt
```

This command objectizes file1.txt at location /ibm/cesSharedRoot/fileset1/Auth_12345/container1/ and enables it for access through the object interface.

For more information, see *mmobj command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related tasks

[Enabling the file-access object capability](#)

Before you can use unified file and object access, you must enable the file-access object capability on the whole cluster.

[Starting and stopping the ibmobjectizer service](#)

The following information provides the commands to start and stop the ibmobjectizer service.

[Setting up the objectizer service interval](#)

Take the following steps to set up the objectizer service interval.

[Enabling and disabling QOS](#)

This topic lists the commands to enable and disable QOS.

[Configuring authentication and setting identity management modes for unified file and object access](#)

You can configure authentication and set the identity management modes for unified file and object access using the following steps.

[Creating or using a unified file and object access storage policy](#)

Use the following steps to create or use a unified file and object access storage policy.

[Associating containers with a unified file and object access storage policy](#)

Use the following steps to associate a container with a unified file and object access storage policy.

[Creating exports on a container that is associated with a unified file and object access storage policy](#)

Use the following steps to create a Network File System (NFS) or Server Message Block (SMB) export on the directory that maps to the container associated with the unified file and object access storage policy.

[Example scenario - administering unified file and object access](#)

The following example describes an end-to-end scenario of administering and configuring unified file and object access.

Example scenario - administering unified file and object access

The following example describes an end-to-end scenario of administering and configuring unified file and object access.

Before you can use the following steps, IBM Storage Scale for object storage must be installed.

This example provides a quick reference of steps that are done for unified file and object access. For more information, see “[Administering unified file and object access](#)” on page 394.

1. Enable the file-access object capability as follows:

```
mmobj file-access enable
```

2. Optional: Change the objectizer service interval as follows:

```
mmobj config change --ccrfile spectrum-scale-objectizer.conf \
--section DEFAULT --property objectization_interval --value 600
```

3. Optional: Change the identity management mode to unified_mode as follows:

```
mmobj config change --ccrfile object-server-sof.conf \
--section DEFAULT --property id_mgmt --value unified_mode
```

4. Optional: Set the ad_domain parameter as follows:

```
mmobj config change --ccrfile object-server-sof.conf \
--section DEFAULT --property ad_domain --value ADDOMAINX
```

5. Create a unified file and object access storage policy as follows:

```
mmobj policy create SwiftOnFileFS --enable-file-access
```

A sample output is as follows.

```
[I] Getting latest configuration from ccr
[I] Creating fileset /dev/gpfs0:obj_SwiftOnFileFS
[I] Creating new unique index and building the object rings
[I] Updating the configuration
[I] Uploading the changed configuration
```

This command also creates a unified file and object access enabled fileset.

6. Create a base container with a unified file and object access storage policy as follows:

```
swift post unified_access -H "X-Storage-Policy: SwiftOnFileFS"
```

7. Store the path that is created for the container by finding it in the newly created fileset as follows:

```
export FILE_EXPORT_PATH=`find /ibm/gpfs0/obj_SwiftOnFileFS/
-name "unified_access"`

# echo $FILE_EXPORT_PATH
/ibm/gpfs0/obj_SwiftOnFileFS/s10041510210z1device1/
AUTH_09271462d54b472c82adecff17217586/unified_access
```

8. Create an SMB share on the path as follows:

```
mmsmb export add unified_access $FILE_EXPORT_PATH
```

A sample output is as follows.

```
mmsmb export add: The SMB export was created successfully
```

9. Create an NFS export on the path:

```
mmnfs export add $FILE_EXPORT_PATH --client \
"*(Access_Type=RW,Squash=no_root_squash,SecType=sys)"
```

A sample output is as follows:

```
192.0.2.2: Redirecting to /bin/systemctl stop nfs-ganesha.service
192.0.2.3: Redirecting to /bin/systemctl stop nfs-ganesha.service
192.0.2.2: Redirecting to /bin/systemctl start nfs-ganesha.service
192.0.2.3: Redirecting to /bin/systemctl start nfs-ganesha.service
NFS Configuration successfully changed. NFS server restarted on all NFS nodes.
```

Note: If it is the first NFS export added to the configuration, the NFS service is restarted on the CES nodes where the NFS server is running. Otherwise, no NFS restart is needed when you add an NFS export.

10. Check the NFS and SMB shares:

```
mmnfs export list
```

A sample output is as follows:

```
Path Delegations Clients
-----
/ibm/gpfs0/obj_SwiftOnFileFS/
s10041510210z1device1/
AUTH_09271462d54b472c82adecff17217586/unified_access none *
mmsmb export list

export          path           guest ok   server smb encrypt
unified_access /ibm/gpfs0/obj_SwiftOnFileFS/
s10041510210z1device1/
AUTH_09271462d54b472c82adecff17217586/unified_access     no        auto

Information:
The following options are not displayed because they do not contain a value:
"browseable"
```

11. Access this export with NFS or SMB clients and create a sample directory and a file:

DirCreatedFromGPFS/File1.txt and DirCreatedFromSMB/File2.txt

You can view the association of ownership when data is created from the SMB interface as follows:

```
ls -l /ibm/gpfs0/obj_SwiftOnFileFS/s10041510210z1device1/
AUTH_09271462d54b472c82adecff17217586/unified_access/DirCreatedFromSMB
total 0
-rwxr--r--. 1 ADDOMAINX\administrator
ADDOMAINX\domain users 20 Oct 21 18:09 File2.txt

mmgetacl /ibm/gpfs0/obj_SwiftOnFileFS/s10041510210z1device1/
AUTH_09271462d54b472c82adecff17217586/unified_access/DirCreatedFromSMB
#NFSv4 ACL
#owner:ADDOMAINX\administrator
#group:ADDOMAINX\domain users
special:owner@:rwx:allow
(X)READ/LIST (X)WRITE/CREATE (X)APPEND/MKDIR (X)SYNCHRONIZE
(X)READ_ACL (X)READ_ATTR (X)READ_NAMED
(-)DELETE (-)DELETE_CHILD (X)CHOWN
(X)EXEC/SEARCH (X)WRITE_ACL (X)WRITE_ATTR (X)WRITE_NAMED

special:group@:r-x-:allow
(X)READ/LIST (-)WRITE/CREATE (-)APPEND/MKDIR (X)SYNCHRONIZE
(X)READ_ACL (X)READ_ATTR (X)READ_NAMED
(-)DELETE (-)DELETE_CHILD (-)CHOWN
(X)EXEC/SEARCH (-)WRITE_ACL (-)WRITE_ATTR (-)WRITE_NAMED

special:everyone@:r-x-:allow
(X)READ/LIST (-)WRITE/CREATE (-)APPEND/MKDIR (X)SYNCHRONIZE
(X)READ_ACL (X)READ_ATTR (X)READ_NAMED
(-)DELETE (-)DELETE_CHILD (-)CHOWN
(X)EXEC/SEARCH (-)WRITE_ACL (-)WRITE_ATTR (-)WRITE_NAMED
```

You can view the container and the file that is created from the REST interface and retention of ownership in the PUT operation as follows:

```
ls -l /ibm/gpfs0/obj_SwiftOnFileFS/s10041510210z1device1/
AUTH_09271462d54b472c82adecff17217586/unified_access/DirCreatedFromSMB/File2.txt

-rwxr--r--. 1 ADDOMAINX\administrator ADDOMAINX\domain users 520038360 Nov 3 11:47
/ibm/gpfs0/obj_SwiftOnFileFS/s10041510210z1device1/AUTH_09271462d54b472c82adecff17217586/
DirCreatedFromSMB/unified_access/File2.txt
```

12. Objectize that file immediately by using the following command or wait for the objectization cycle to complete:

```
mmobj file-access objectize --object-path \
/ibm/gpfs0/obj_SwiftOnFileFS/s10041510210z1device1/AUTH_09271462d54b472c82adecff17217586/unified_access/
File2.txt
```

13. List the contents of the container by using the Swift client that is configured with all variables as follows:

```
swift list unified_access
```

A sample output is as follows:

```
DirCreatedFromGPFS/File1.txt  
DirCreatedFromSMB/File2.txt
```

14. Download that object by using the Swift client that is configured with all variables as follows:

```
swift download unified_access/File2.txt
```

Note: The steps that are done by using Swift commands can also be done by using **curl**. For more information, see “[curl commands for unified file and object access related user tasks](#)” on page 411.

Related tasks

[Enabling the file-access object capability](#)

Before you can use unified file and object access, you must enable the file-access object capability on the whole cluster.

[Starting and stopping the ibmobjectizer service](#)

The following information provides the commands to start and stop the ibmobjectizer service.

[Setting up the objectizer service interval](#)

Take the following steps to set up the objectizer service interval.

[Enabling and disabling QOS](#)

This topic lists the commands to enable and disable QOS.

[Configuring authentication and setting identity management modes for unified file and object access](#)

You can configure authentication and set the identity management modes for unified file and object access using the following steps.

[Creating or using a unified file and object access storage policy](#)

Use the following steps to create or use a unified file and object access storage policy.

[Associating containers with a unified file and object access storage policy](#)

Use the following steps to associate a container with a unified file and object access storage policy.

[Creating exports on a container that is associated with a unified file and object access storage policy](#)

Use the following steps to create a Network File System (NFS) or Server Message Block (SMB) export on the directory that maps to the container associated with the unified file and object access storage policy.

[Enabling object access for selected files](#)

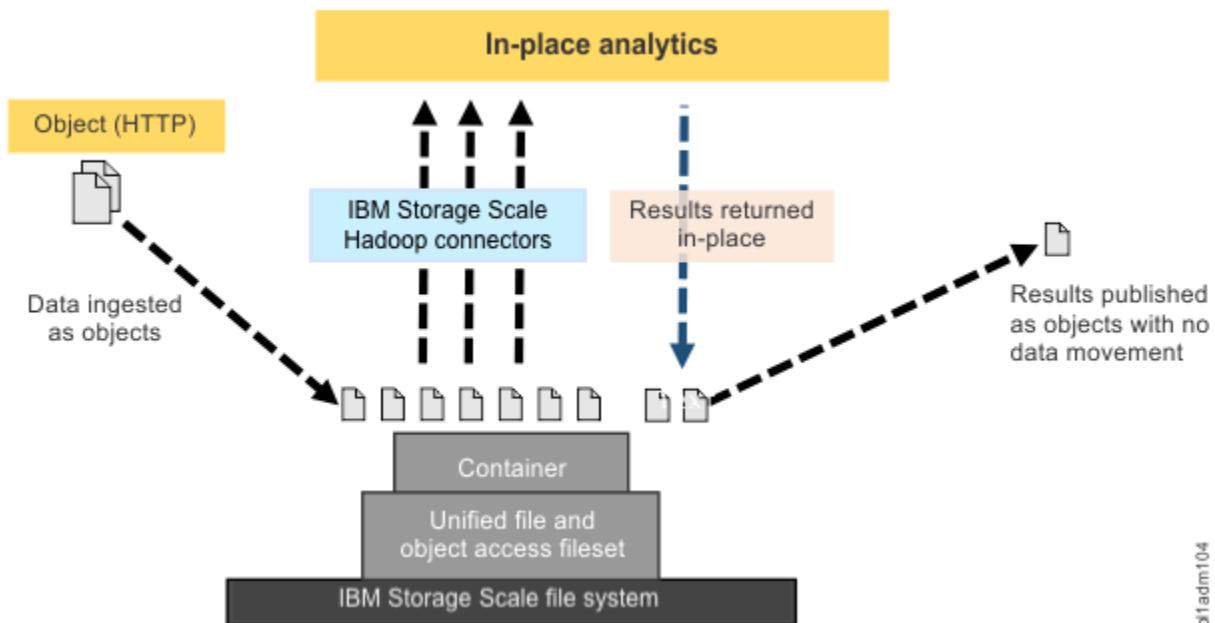
Use the following steps to objectize files under all containers that are associated with the unified file and object access storage policy under a specified account.

In-place analytics using unified file and object access

Use the following information to use in-place object data analytics by using unified file and object access.

Unified file and object access is one of the key features of IBM Storage Scale for object storage that enables direct object access as files from the traditional file access such as POSIX, NFS, or SMB and vice versa. Using object storage policies for containers you can ingest in IBM Storage Scale for object storage be accessed as files as well as allow files ingested by using file protocols available for object access. This feature enables data analytics of object data that is hosted on IBM Storage Scale, where you can use in-place object data analytics. IBM Storage Scale supports Hadoop connectors that you can use to run analytics on the object data that is accessible from the file interface and generate in-place results that are directly accessible from the object interface. This prevents any movement of data across object interfaces and thus proves to be a suitable platform for object storage and integrated in-place analytics for the data hosted by it.

The following diagram shows an IBM Storage Scale object store with unified file and object access. The object data is available as file on the same filesset. IBM Storage Scale Hadoop connectors allow the data to be directly used for analytics.



b1adm104

Figure 12. In-place analytics with unified file and object access

Limitations of unified file and object access

Read about the limitations to unified file and object access in IBM Storage Scale.

The following limitations apply:

- The base container must be created from the object interface in the filesset that is being used for the unified access storage policy. Then, only the files that are added after that are enabled for object access.
- Concurrent access to the same object or file from file and object interface at the same time leads to an undefined state. You can prevent conflicts:
 1. You can have your workflow enforce the limitation.
 2. You can explicitly enforce read-only access for some periods. With NFS and SMB, it can be done in the export definition. With POSIX, it can be done by using ACLs.
- Files or directories that are created at the base container level cannot be enabled for object access. Only the files that are created under the container are enabled for object access.
- Multi-region object deployment cannot be used with unified file and object access.
- Object versioning is not supported by unified file and object access.
- Special files such as device files and pipes, file clones, and soft links can exist in the object container directory, but they are not visible from the object interface.
- Containers must be deleted from the object interface. Container directories that are deleted from the file interface continue to show up in the container listing until the container is deleted from the object interface.
- GPFS quota and Swift container and account quota are mutually exclusive in IBM Storage Scale 4.2 and later. The user quota that is assigned to a user or a group in GPFS does not relate to the container quota defined in the object interface.

- Swift large object support (dynamic large object and static large object) is not available with unified file and object-enabled containers. S3 multipart uploads are also not supported by unified file and object-enabled containers.
- GPFS immutability is not supported by unified file and object access.
- Only object metadata can be viewed and modified from the object interface. Extended attributes that are defined from the file interface cannot be viewed from the object interface.
- Empty directories that are created from the file interface within a container are not objectized and are not listed in the container listing.
- Files or directories with " :: " or newline characters in their names are not supported. These files and data that resides in these containers are not objectized.
- Change of authentication scheme of file or object might directly impact access to existing file or object data. Therefore, change of authentication is not supported as it results in loss of access for the users to the existing data on the system.
- Object ETag is inaccurate in the following scenarios:
 - Whenever an object is modified from the file interface.
 - If the **user.swift.metadata** extended attribute is explicitly deleted from the file interface, ETag is not present because of which the headers do not return correct results. You must wait for at least one cycle of objectization or explicitly objectize that file to use the ETag conditional request feature.

Note: An incorrect ETag is corrected when a GET or HEAD request is done on the object.

- The IBM Storage Scale ILM policy rules work with file-extended attributes, and rules can be easily created based on extended attributes and their values. However, these rules do not work directly over Swift user-defined metadata. All of Swift user-defined metadata is stored in a single extended attribute in the IBM Storage Scale file system. To create ILM rules, the format and sequence in which the attributes are stored must be noted. Rules can then be created by constructing wildcard-based filters.
- SELinux must be in the Permissive or Disabled mode to enable object access for the existing filesets.
- The conditional request, such as If-Match and If-None-Match when used with swift or curl client that does ETag comparison does not work for the existing data that is enabled for object access by using the **mmobj file-access link-fileset** command. If the --update-listing option is used, the feature can be used after the objectizer service interval.
- The swift and curl clients might report successful container deletion after a delete operation is triggered on a container that contains linked filesets. So, the directory corresponding to the container and the symlinks of the linked filesets are not deleted and must be deleted manually.
- The swift COPY API (when used on linked fileset) does not copy the object metadata. Use the swift POST API instead.
- The performance of the objectization process is impacted by the size and the concurrent usage of the associated containers. Because the objectization process compares the files in the fileset to the objects in the container databases, performance degrades as the containers get larger or if the container usage is high during the objectization process.

Constraints applicable to unified file and object access

The following constraints are applicable while creating and accessing objects and containers for unified file and object access:

- The name of the container can be no more than 255 characters.
- The name of the object can be no more than 214 characters.
- The path name of the object must not include successive forward slashes.
- The name of the container and the object must not be a single period (.) or a double period (..). However, a single period or a double period can be part of the name of the container and the object.

The system returns the following error message when the constraints are not met:

The swift constraints listed in the following table are also applicable to unified file and object access.

<i>Table 25. Configuration options for [swift-constraints] in swift.conf</i>	
Option	Limit
MAX_FILE_SIZE	5497558138880 (5 TiB)
MAX_META_NAME_LENGTH	128
MAX_META_VALUE_LENGTH	256
MAX_META_COUNT	90
MAX_META_OVERALL_SIZE	4096
MAX_HEADER_SIZE	8192
CONTAINER_LISTING_LIMIT	10000
ACCOUNT_LISTING_LIMIT	10000
MAX_ACCOUNT_NAME_LENGTH	256
VALID_API_VERSIONS	["v1", "v1.0"]
EXTRA_HEADER_COUNT	0

Note: These values can be changed by using the following command for the `swift.conf` file in `swift-constraints` section:

```
mmobj config change
```

Data ingestion examples

Use the following example steps for data ingestion in the following scenarios.

You must consider data ingestion and access:

- You can have data ingestion through object interface and access through file interface.
 - You can have data ingestion through file interface and access through object interface.
 - You can have data ingestion and access through object and file interfaces concurrently.
1. This step is for the standard REST client. Get proper authentication token from the Authentication URL by using proper credentials to authorize on further requests.
 2. This step is for the standard REST client. Use the token that is obtained in the previous step to do the PUT, POST, DELETE, COPY (object only), or HEAD operations for objects that are under container that is created with unified file and object access storage policy.
 3. This step is for the standard file client step. Mount SMB or NFS exports on respective NFS or SMB clients with regular mount commands or interface available with file clients:

```
mount -t cifs -o username=STORAGE5TEST\\fileuser1,password=Passw0rd5,vers=3.0 //192.0.2.4/unified_access /mnt/unified_access
```

For the following data ingestion example steps that are done by using the `curl` command, this setup is assumed:

- The user is "fileuser".
- The password is "Password6".
- The account name is "admin".
- The host is `specscaleswift.example.com`.

1. Run the following command to obtain the authentication token:

```
curl -s -i -H "Content-Type: application/json"
-d '{"auth": {"identity": {"methods": ["password"], "password": {"user": {"name": "fileuser", "domain": {"name": "Default"}, "password": "Passw0rd6"}}, "scope": {"project": {"name": "admin", "domain": {"name": "Default"}}}}}'
http://specscsaleswift.example.com:35357/v3/auth/tokens
```

The auth token that is obtained in this step must be stored in the `$AUTH_TOKEN` variable.

2. Run the following command to obtain the project list:

```
curl -s -H "X-Auth-Token: $AUTH_TOKEN" http://specscsaleswift.example.com:35357/v3/projects
```

The project ID obtained in this step must be stored in the `$AUTH_ID` variable.

3. Run the following command to do a PUT operation:

```
curl -i -s -X PUT --data @/tmp/file.txt -H "X-Auth-Token: $AUTH_TOKEN" "http://specscsaleswift.example.com:8080/v1/AUTH_$AUTH_ID/RootLevelContainer/TestObj.txt"
```

This command uploads the `/tmp/file.txt` file.

4. Run the following command to set up the metadata age of the uploaded object:

```
curl -i -s -X POST -H "X-Auth-Token: $AUTH_TOKEN" -H "X-Container-Meta-Age:21" http://specscsaleswift.example.com:8080/v1/AUTH_$AUTH_ID/RootLevelContainer/TestObj.txt
```

5. Run the following command to read the metadata:

```
curl -i -s --head -H "X-Auth-Token: $AUTH_TOKEN" http://specscsaleswift.example.com:8080/v1/AUTH_$AUTH_ID/RootLevelContainer/TestObj.txt
```

curl commands for unified file and object access related user tasks

Use the following curl commands for user tasks that are related to unified file and object access.

For the following commands, it is assumed that:

- A token is generated and it is exported as an environment variable `AUTH_TOKEN`.
- A swift endpoint URL for the project (tenant) for which token is generated.
- A unified file and object access storage policy that is named `SwiftOnFileFS` is already created.

1. Create a container that is named `unified_access` with unified file and object access storage policy by running the `curl` command as follows:

```
curl -v -i -H "X-Auth-Token: $AUTH_TOKEN"
-X PUT http://specscsaleswift.example.com:8080/v1/AUTH_cd1a29013b6842939a959dbda95835df/
unified_access/
-H "X-Storage-Policy: SwiftOnFileFS"
```

In this command, `http://specscsaleswift.example.com:8080/v1/AUTH_cd1a29013b6842939a959dbda95835df/` is the endpoint URL for a project (tenant) by using the container `unified_access` that is created with `SwiftOnFileFS` as the storage policy.

2. Upload an object in the container that is associated with the unified file and object access storage policy by running the `curl` command as follows:

```
curl -v -i -H "X-Auth-Token: $AUTH_TOKEN"
-X PUT http://specscsaleswift.example.com:8080/v1/AUTH_cd1a29013b6842939a959dbda95835df
/unified_access/object1
--data-binary @imageA.jpg
```

- Download the object that is in the unified file and object access container by running the **curl** command as follows:

```
curl -v -i -H "X-Auth-Token: $AUTH_TOKEN"
-X GET http://specscsaleswift.example.com:8080/v1/AUTH_cd1a29013b6842939a959dbda95835df
/unified_access/samplefile.txt
```

- List the contents of the unified file and object access container by running the **curl** command as follows:

```
curl -v -i -H "X-Auth-Token: $AUTH_TOKEN"
-X GET http://specscsaleswift.example.com:8080/v1/AUTH_cd1a29013b6842939a959dbda95835df
/unified_access/
```

Configuration files for IBM Storage Scale for object storage

Use the following information to manage options in configuration files that are used for IBM Storage Scale for object storage that includes the unified file and object access feature. These configuration files are located in the /etc/swift directory.

For more information, see [“Changing options in configuration files” on page 415](#).

object-server-sof.conf file

This file contains identity management modes for unified file and object access (**id_mgmt**). This file contains AD domain name (**ad_domain**) if AD is configured.

Table 26. Configurable options for [DEFAULT] in object-server-sof.conf	
Configuration option = Default value	Description
id_mgmt = local_mode	Defines the object server behavior while it assigns user or group ownership to newly created objects, when the objects are accessed by using the file interface. The allowed values are <code>local_mode</code> and <code>unified_mode</code> With <code>local_mode</code> , the new objects are owned by the swift user. In <code>unified_mode</code> , the identity of the user that makes the PUT request is fetched from the configured directory server.
ad_domain	When using Active Directory (AD), defines the AD domain from which the user identity must be fetched when object server is operating in the <code>unified_mode</code> identity management mode. Note: When you clean up object authentication, you must manually remove this entry. For more information, see “Configuring authentication and setting identity management modes for unified file and object access” on page 398 .
tempfile_prefix = .ibmtmp_	Indicates the prefix to be used for the temporary file that is created when a file uploaded.
disable_fallocate = true	Overrides the default swift allocate behavior, and relies on the GPFS allocate features, excludes 'fast fail' checks.
disk_chunk_size = 65536	Indicates the size of chunks to read or write to disk (needs be equal to the file system block size).

Table 26. Configurable options for [DEFAULT] in object-server-sof.conf (continued)

Configuration option = Default value	Description
network_chunk_size = 65536	Indicates the size of chunks to read or write over the network (needs to be equal to the file system block size).
log_statsd_host = localhost	If it is not set, the StatsD feature is disabled.
log_statsd_port = 8125	Indicates the port number for the StatsD server.
log_statsd_default_sample_rate = 1.0	Defines the probability of sending a sample for any specified event or timing measurement.
log_statsd_sample_rate_factor = 1.0	It is not recommended to set it to a value less than 1.0. If the frequency of logging is too high, tune the log_statsd_default_sample_rate instead.
log_statsd_metric_prefix =	Indicates the prefix that is added to every metric sent to the StatsD server.
retain_acl = yes	Indicates whether to copy the ACL from an existing object. Allowed values are yes or no.
retain_winattr = yes	Indicates whether to copy the Windows attributes from an existing object. Allowed values are yes or no.
retain_xattr = yes	Indicates whether to copy the extended attributes for the user namespace from an existing object. Allowed values are yes or no.
retain_owner = yes	Indicates whether to copy the UID/GID owners from an existing object. Allowed values are yes or no.

Note: Files with the .ibmtmp prefix or the one configured in the object-server-sof.conf configuration file are not objectized.

When you set the **retain_*** options to yes, the following attributes are retained:

- The extended attributes in the user namespace except for the **user.swift.metadata** key that contains swift metadata and it is expected to be new.
- Windows attributes

When you set the **retain_*** options to yes, the following attributes are not retained:

- Extended attributes in system, security, and trusted namespaces.

Note: These attributes are not retained in an object's copy object operation also.

Retaining ACLs, Windows attributes, file extended attributes, and ownership, when an object is PUT over an existing object in unified file and object access enabled containers depends on your specific use case and your discretion. For example, if you are using object and file access to refer to the same data content in such a way that the object protocol might completely replace the data content in such that it might be new content from the file interface as well, then you might choose to not retain the existing file ACL and extended attributes. For such a use case, you might change the default values to not retain the file ACLs, extended attributes, and ownership.

Note: If you are unsure about whether to retain these attributes or not, you might want to use the default values of retaining ACLs, Windows attributes, file extended attributes, and ownership. The default values in this case are more aligned with the expected behavior in a multiprotocol setup.

spectrum-scale-object.conf file

This file contains cluster or fileset configuration information. This file is unique to a site.

Table 27. Configurable options for [capabilities] in spectrum-scale-object.conf

Configuration option = Default value	Description
file-access-enabled = false	Indicates the state for the file-access capability. It can be either true or false.
multi-region-enabled = true	Indicates the state for the multi-region capability. This option cannot be changed.
s3-enabled = true	Indicates the state for the s3 capability. This option cannot be changed.

spectrum-scale-objectizer.conf file

- Contains the ibmobjectizer service configuration information

Table 28. Configuration options for [DEFAULT] in spectrum-scale-objectizer.conf

Configuration option = Default value	Description
objectization_tmp_dir	Indicates the temporary directory to be used by ibmobjectizer. The directory must be a path on any GPFS file system. The default value is autofs with the path of the base file system for object.
objectization_threads = 24	Indicates the maximum number of threads that ibmobjectizer creates on a node.
batch_size = 100	Indicates the maximum number of files that ibmobjectizer process in a thread.
objectization_interval = 1800	Indicates the time interval, in seconds, between the completion of an objectization cycle and the beginning of the next objectization cycle.
connection_timeout = 25	Indicates the connection timeout for an account, container, or object server request from ibmobjectizer,
response_timeout = 25	Indicates the response timeout for an account, container, object server request from ibmobjectizer
qos_iops_target = 0	Indicates the value that is assigned to ibmobjectizer to limit its resource usage. Value is given in IOPS unit, such as - 100, 400, 0. 0 means infinite.

Table 29. Configuration options for [IBMOBJECTIZER-LOGGER] in spectrum-scale-objectizer.conf

Configuration option = Default value	Description
log_level = INFO	Indicates the logging level. The allowed value is INFO, DEBUG, WARN, or ERROR.

object-server.conf file

This file is used to set Swift timeout values on the lock_path calls to handle GPFS delays better.

Table 30. Configuration options for object-server.conf

Configuration option = Default value	Description
partition_lock_timeout = 10	The timeout value while the object server tries to acquire a lock on the partition path during object create, update, and delete processes. The default value is 10.

/etc/sysconfig/memcached file

- Used to improve the performance of the internal lookups in the framework

Table 31. Configuration options for /etc/sysconfig/memcached

Configuration option = Default value	Description
MAXCONN = 4096	The value is set to 4096 unless the current value is higher than 4096.
CACHESIZE = 2048	The value is set to 2048 unless the current value is higher than 2048.

proxy-server.conf file

This file is used to improve the performance of the internal lookups in the framework

Table 32. Configuration options for proxy-server.conf

Configuration option = Default value	Description
memcache_max_connections = 8	The default value is set to 8.
memcache_servers	This parameter is dynamically set to the nodes that are running the object protocol for memcache to work in a clustered environment.

Changing options in configuration files

You can use the following commands to change the values of the options in the configuration files.

- Change the value of an option in the [DEFAULT] section of the object-server-sof.conf file as follows:

```
mmobj config change --ccrfile object-server-sof.conf --section DEFAULT  
--property OPTIONNAME --value NEWVALUE
```

- Change the value of an option in the [IBMOBJECTIZER-LOGGER] section of the spectrum-scale-objectizer.conf file as follows:

```
mmobj config change --ccrfile spectrum-scale-objectizer.conf --section IBMOBJECTIZER-LOGGER  
--property OPTIONNAME --value NEWVALUE
```

Note: Only some options are configurable. If an option cannot be changed, it is mentioned in the respective option description.



Attention: When a configuration file is changed by using these commands, it takes several seconds for the changes to be synchronized across the whole cluster (depending on the size of the cluster). Therefore, when you run multiple commands to change configuration files, you must plan for an adequate time interval between the execution of these commands.

Backing up and restoring object storage

Snapshots are a good way to protect data from various errors and failures. Moving them to a separate backup storage system can provide better protection against catastrophic failures of the entire storage system and might even allow the data to be stored at a lower cost. The following information describes the manual steps that are needed to back up and restore the object storage and its configuration information.

In the examples, the steps to back up the Keystone configuration files and database are not given because backing up the Keystone configuration files and database is the user's responsibility. You can use OpenStack backup procedures for this task. For more information, see [Chapter 14. Backup and Recovery](#).

Note:

- The same version of the IBM Storage Protect backup-archive client needs to be installed on any nodes that are running the **mmbbackup** command.

Backing up the object storage

IBM Storage Scale Object Nodes and IBM Storage Protect client nodes need to be available with the object file system mounted on each node when the backup is being created. The IBM Storage Protect server needs to also be available.

Store any relevant cluster and file system configuration data in a safe location outside your GPFS cluster environment. This data is essential to restoring your object storage quickly, so you might want to store it in a site in a different geographical location for added safety.

Follow these steps to back up the object storage manually:

Remember: The sample file system used throughout this procedure is called **smallfs**. Replace this value with your file system name wherever necessary.

1. Back up the cluster configuration information.

The cluster configuration needs to be backed up by the administrator. The following cluster configuration information is necessary for the backup:

- IP addresses are needed.
- Node names are needed.
- Roles are needed.
- Quorum and server roles are needed.
- Cluster-wide configuration settings from the **mmchconfig** command are needed.
- Cluster manager node roles are needed.
- Remote shell configuration is needed.
- Mutual Secure Shell (SSH) and Remote Shell (RSH) authentication setup are needed.
- Cluster UID is needed.

Note: Comprehensive configuration information can be found in the **mmsdrfs** file.

2. Preserve disk configuration information.

Disk configuration needs to also be preserved to recover a file system. The fundamental disk configuration information needed for a backup intended for disaster recovery is as follows:

- The number of disk volumes that were previously available is needed.
- The sizes of those volumes are needed.

Important: To recover from a total file system loss, at least as much disk space as was previously available is needed for restoration.

It is only possible to restore the image of a file system onto replacement disks if the disk volumes available are of similar enough sizes to the originals. This allows any data to be restored to the new disks. The following disk configuration information is necessary for the recovery:

- Disk device names are needed.
- Disk device sizes are needed.
- The number of disk volumes is needed.
- NSD server configuration is needed.
- Disk RAID configurations are needed.
- Failure group designations are needed.
- The `mmsdrfs` file contents are needed.

3. Back up the GPFS™ file system configuration information.

In addition to the disks, the file system built on those disks has the following configuration information that can be captured using the `mmbackupconfig` command:

- Block size can be captured.
- Replication factors can be captured.
- Number and size of disks can be captured.
- Storage pool layout can be captured.
- Filesets and junction points can be captured.
- Policy rules can be captured.
- Quota information can be captured.
- Other file system attributes can be captured.

The file system configuration information can be backed up into a single file using a command similar to the following:

```
mmbackupconfig smallfs -o /tmp/smallfs.bkpcfg.out925
```

4. Save the following IBM Storage Protect configuration files for each IBM Storage Protect client node in the same safe location outside of your GPFS cluster.

`/etc/adsm/TSM.PWD`

Contains the client password that is needed to access IBM Storage Protect. This file is present only when the IBM Storage Protect server setting of authentication is set to on.

`/opt/tivoli/tsm/client/ba/bin/dsm.sys` and

`/opt/tivoli/tsm/client/ba/bin/dsm.opt`

Contains the IBM Storage Protect client configuration files.

5. Back up the object storage content to an IBM Storage Protect server by running the `mmbackup` command:

- a) Create a global snapshot by running the following command:

```
mmcsnapshot <file system device> <snapshot name>
```

For example, create a snapshot that is named `objects_globalsnap1` by running the following command:

```
mmcsnapshot smallfs objects_globalsnap1
```

- b) Create global and local work directories by running the following commands:

```
mkdir -p /smallfs0/.es/mmbackupglobal
```

```
mkdir -p /smallfs0/.es/mmbackupslocal
```

- c) Run the following command to start the snapshot-based backup:

```
mmbackup <file system device> -t incremental -N <TSM client nodes> \ -g <global work directory> \ -s <local work directory> \ -S <global snapshot name> --tsm-servers <tsm server> --noquote
```

The \ indicates the line wrap:

```
mmbackup smallfs -t incremental -N node1,node2 \
-g /smallfs0/.es/mmbackupglobal \
-s /smallfs0/.es/mmbackuplocal \
-S objects_globalsnap1 --tsm-servers tsm1 --noquote
```

In this example:

-N

Specifies the nodes that are involved in the backup process. These nodes need to be configured for the IBM Storage Protect server that is being used.

-S

Specifies the global snapshot name to be used for the backup.

--tsm-servers

Specifies which IBM Storage Protect server is used as the backup target, as specified in the IBM Storage Protect client configuration dsm.sys file.

There are several other parameters available for the **mmbackup** command that influence the backup process, and the speed with which its handles the system load. For example, you can increase the number of backup threads per node by using the **-m** parameter. For the full list of parameters available, see the *mmbackup command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

- d) Run the following command to remove the snapshot that was created in step 6a:

```
mmdelsnapshot <file system device> <snapshot name>
```

You can use the following example:

```
mmdelsnapshot smallfs objects_globalsnap1
```

Restoring the object storage

You need to meet the following prerequisites before beginning the recovery procedure:

1. Restore the GPFS cluster with the same node names that were used during the backup procedure
2. Restore your OpenStack Keystone server and make sure that it is operational.
3. Install Swift software on any IBM Storage Scale object nodes.
4. Install the IBM Storage Protect backup-archive client software on the IBM Storage Scale object nodes that were clients previously.

Note: IBM Storage Scale object nodes and IBM Storage Protect client nodes need to be available when the object storage configuration and contents are being restored.

After you perform the prerequisite procedures, you can begin the recovery procedure.

Note: The sample file system that is used throughout this procedure is called **smallfs**. Replace this value with your file system name wherever necessary.

1. Retrieve the base file system configuration information.

Use the following command to generate a configuration file that contains the details of the former file system:

```
mmrestoreconfig smallfs -i /tmp/smallfs.bkpcfg.out925 -F
smallfsQueryResultFile
```

2. Re-create the NSDs when they are missing.

Using the output file that is generated in the previous step as a guide, the administrator might need to re-create NSD devices for use with the restored file system. In the output file, the NSD configuration section contains the NSD information:

```
##### NSD configuration #####
## Disk descriptor format for the mmcrlnsd command.
## Please edit the disk and desired name fields to match
## your current hardware settings.
##
```

```

## The user then can uncomment the descriptor lines and
## use this file as input to the -F option.
#
# %nsd:
#   device=DiskName
#   nsd=nsd8
#   usage=dataAndMetadata
#   failureGroup=-1
#   pool=system
#

```

If changes are needed, edit the file in a text editor and follow the included instructions to use it as input for the **mmcrnsd** command and run the following command:

```
mmcrnsd -F StanzaFile
```

3. Re-create the base file system.

The administrator needs to re-create the initial file system. The output query file created in step 1 can be used as a guide. The following example shows the section of this file that is needed when re-creating the file system:

```

##### File system configuration #####
## The user can use the predefined options/option values
## when recreating the file system. The option values
## represent values from the backed up file system.
#
# mmcrfs FS_NAME NSD_DISKS -j cluster -k posix -Q yes -L 4194304 --disable-fastea
-T /smallfs -A no --inode-limit 278016#

```

4. Restore the essential file system configuration.

The essential file system configuration can be restored to the file system that was created in the previous step by running the **mmrestoreconfig** command:

```
mmrestoreconfig smallfs -i /tmp/smallfs.bkpcfg.out925
```

5. Run the following command to mount the object file system on all of the nodes:

mmount <file system device> -a

For example, mount the file system with the following command:

```
mmount smallfs -a
```

6. Restore the configuration of the IBM Storage Protect client nodes by copying the saved configuration files from their saved location to each IBM Storage Protect client node.

- The IBM Storage Protect client config files dsm.opt and dsm.sys needs to be restored to /opt/tivoli/tsm/client/ba/bin/.
- If the IBM Storage Protect client password file, TSM.PWD, is saved during the backup procedure, it needs to be restored to /etc/adsm/.
- Run the following command to verify that each IBM Storage Protect client node can communicate with the IBM Storage Protect server without prompting for a password: **dsmc q sess**

7. Restore the object storage data from the IBM Storage Protect server.

- Run the **dsmc restore** command as shown to start a no-query restore on an IBM Storage Protect client node.

```
dsmc restore <GPFS Object path> -subdir=yes -disablensqr=no \
-servername=<tsm server> -errorlogname=<error log path>.
```

You can use the following example:

```
dsmc restore /smallfs/ -disablensqr=no \
-servername=tsm1 -errorlogname=/tmp/object_restore.log
```

- When the restore jobs are completed, check the error logs. If any errors are found, correct them so that the restore operations finish successfully.

Improving recovery time

The **dsmc restore** command starts a single restore job on a single node. This job might need a long period to restore any object data. To improve the restore performance, start separate restore jobs on different IBM Storage Protect client nodes.

You can create separate restore jobs by splitting a single restore task into several smaller ones. One way to create separate restore jobs is to specify the restore path for the object data that is deeper in the IBM Storage Scale object path.

For example, instead of starting the restore with the root of the IBM Storage Scale object path, start the object restore at the virtual devices level. If you have 40 virtual devices that are configured, you might start 40 independent restore jobs to restore the object data. Then, distribute the jobs to the different IBM Storage Protect client nodes. Additionally, you start a single restore job for any files under the account and container path.

With this approach, care needs to be taken not to overload the IBM Storage Protect client nodes or the IBM Storage Protect server. You might want to experiment to determine the most optimal mix of jobs.

For example, if there are four IBM Storage Scale object nodes, each with the IBM Storage Protect client installed and configured, you might use the following types of commands:

1. On the first IBM Storage Scale object node, run a restore job for each of the first 10 virtual devices by running the following commands:

```
dsmc restore /gpfs0/objectfs/o/z1device0 -subdir=yes -disablencqr=no \
-servername=tsm1
dsmc restore /gpfs0/objectfs/o/z1device1 -subdir=yes -disablencqr=no \
-servername=tsm1
#<repeat for z1device2 - z1device9>
```

2. On the second node, run a restore job for each of the next 10 virtual devices. Continue the pattern on the remaining IBM Storage Scale object nodes so that any virtual devices under the o subdirectory are restored. Also, start a single restore job for any account and container data under the ac subdirectory by running the following command:

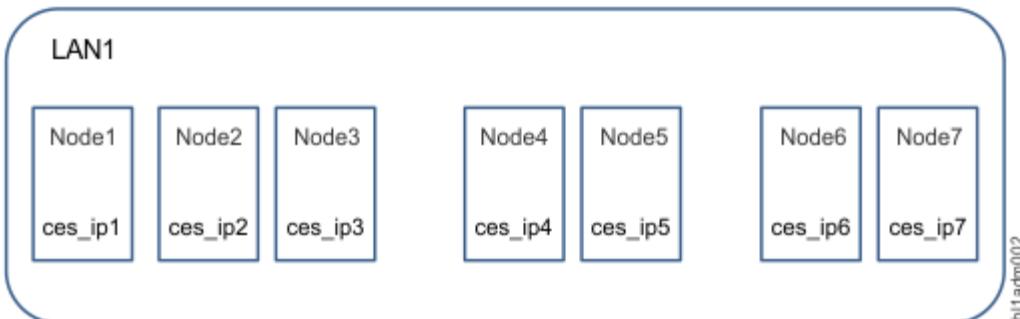
```
dsmc restore /gpfs0/objectfs/ac -subdir=yes -disablencqr=no -servername=tsm1
```

The most efficient restore approach depends on many factors, including the number of tape drives, IBM Storage Protect client configuration, and network bandwidth. You might need to experiment with your configuration to determine the most optimal restore strategy.

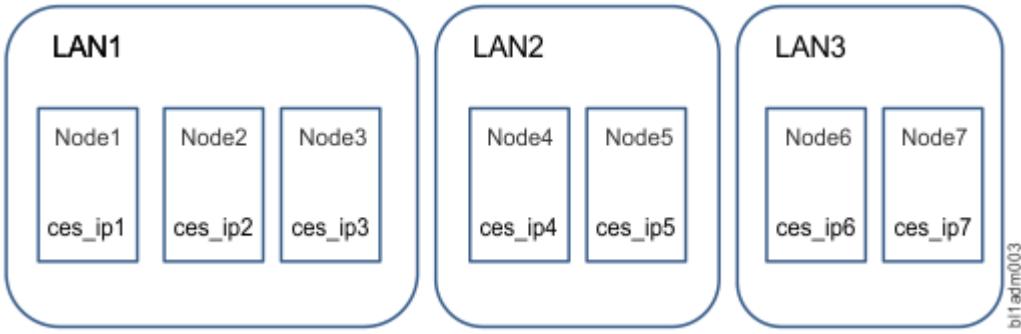
Configuration of object for isolated node and network groups

You can configure object for isolated node and network groups.

Object needs constant network access between all the configured Cluster Export Services (CES) IP addresses. The standard configuration uses all the available CES IP addresses.



Note: If a cluster configuration has an isolated node and network groups and CES IP addresses have been assigned to those groups, parts of the object store are not accessible.



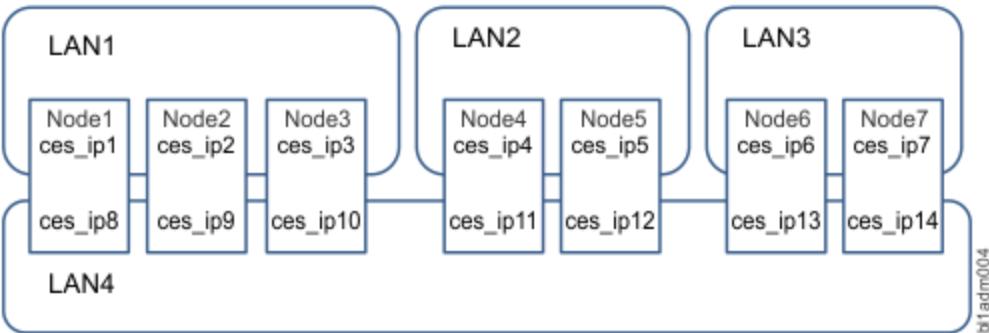
In this configuration, a network and node group that must be used for object can be configured in the `spectrum-scale-object.conf` file. Only the CES IP addresses of this group are used for object.

Note: Only one object group can be used.

Note: If the Singleton and database attributes of the IP address assignments are changed manually by using the `mmces address change` command, only the IP addresses that belong to the object group can be used.

Configuration example

In the following example, LAN1 is used as the object group, IP1, IP2, and IP3 are used for object. And, the object store is fully available. If only LAN1 is used, the used object services will be limited to Node1, Node2, and Node3. To distribute the object load to all the nodes in the cluster, create a network that spans across all nodes. Create an object group and assign all nodes to it. Add new CES IP addresses, at least one CES IP address per protocol node. Then, assign the IP addresses to the same Object Group.



1. To set up the object group and create the LAN4 group by adding nodes to groups, run the following command:

```
mmchnode --ces-group=LAN4 -N Node1,Node2,Node3,Node4,Node5,Node6,Node7
```

Run the following command to add CES IP addresses to the group:

```
mmces address add --ces-ip ces_ip8,ces_ip9,ces_ip10,ces_ip11,
ces_ip12,ces_ip13,ces_ip14 --ces-group LAN4T
```

Run the following command to move the existing CES IP addresses to the group:

```
mmces address change --ces-ip ces_ip8,ces_ip9,ces_ip10,ces_ip11,
ces_ip12,ces_ip13,ces_ip14 --ces-group LAN4
```

2. To set up the object group when object has already been configured, run the following command. The following command is on one line:

```
mmobj config change --ccrfile spectrum-scale-object.conf --section node-group
--property object-node-group --value LAN4
```

To synchronize the ring files, run the following command:

```
/usr/lpp/mmfs/bin/mmcesobjcrring --sync
```

3. To set up the object group when object has not been configured, use the --ces-group option of the **mmobj swift base** command:

```
mmobj swift base -g /gpfs/ObjectFS --cluster-hostname cluster-ces-ip.ibm --local-keystone  
--enable-s3 --pwd-file mmobjpwd --ces-group LAN4
```

CES IP addresses ranging from ces_ip8 to ces_ip14 are used by the object store and the object load is distributed to all the nodes in the cluster. These IP addresses can be used for client connections. These IP addresses can also be used for traffic between the Swift proxy service and the account, container, and object services.

Enabling the object heatmap policy

Use this procedure to enable the object heatmap policy.

Understand how to use a file heat policy.

1. Run the following command to create a file that is named **file_heat_policy** and also add the following policy:

```
RULE 'DefineTiers' GROUP POOL 'TIERS'  
IS 'system' LIMIT(70)  
THEN 'gold' LIMIT(75)  
THEN 'silver'  
RULE 'Rebalance'  
MIGRATE FROM POOL 'TIERS'  
TO POOL 'TIERS' WEIGHT(FILE_HEAT)  
FOR FILESET('Object_Filesset')  
WHERE NAME LIKE '%.data'
```

This policy places the most frequently accessed objects in the SSD-backed system pool until the system pool reaches 70% of its capacity usage. Frequently accessed objects are placed in the gold pool until it reaches 75% capacity usage.

Note:

The temporary files that are generated by Swift are not moved between storage tiers because they are all eventually replaced with permanent files that have the .data extension. Moving temporary files to system, gold, or silver storage pools results in unnecessary data movement.

2. To enable the object heatmap policy for unified file and object access, identify the filename prefix for temporary files that are created by Swift in unified file and object access. The file name prefix is configured in the **object-server-sof.conf** directory. Run the following command to fetch the file name prefix:

```
grep tempfile_prefix /etc/swift/object-server-sof.conf  
tempfile_prefix = .ibmtmp_
```

3. Run the following command to determine the filesets that are enabled for unified file and object access:

```
mmobj policy list  
Index Name Default Deprecated Fileset Functions  
-----  
0 SwiftDefault yes obj_fset  
1317160 Sof obj_Sof file-and-object-access
```

4. Run the following command to create a heat-based migration rule by creating the following file:

```
RULE 'DefineTiers' GROUP POOL 'TIERS'  
IS 'system' LIMIT(70)  
THEN 'gold' LIMIT(75)  
THEN 'silver'  
RULE 'Rebalance' MIGRATE FROM POOL 'TIERS'
```

```

TO POOL 'TIERS' WEIGHT(FILE_HEAT)
FOR FILESET('obj_Sof')
WHERE NAME NOT LIKE '.ibmtmp_%'

```

Note:

- The fileset name is derived from Step 3. Multiple fileset names can be separated by comma.
- The filename prefix in the WHERE clause is derived from Step 2. By using this filter, the migration of temporary files is skipped - which avoids unnecessary data movement.

5. Run the following command to test the policy:

```

mmapplypolicy fs1 -P object_heat_policy -I test
[I] GPFS Current Data Pool Utilization in KB and %
Pool_Name          KB_Occupied      KB_Total    Percent_Occupied
gold               169462784       6836715520   2.478716330%
silver              136192        13673431040   0.000996034%
system              8990720        13673431040   0.065753211%
[I] 6050 of 42706176 inodes used: 0.014167%. [I] Loaded policy rules from object_heat_policy.
Evaluating policy rules with CURRENT_TIMESTAMP = 2015-11-22@02:30:19 UTC
Parsed 2 policy rules.
RULE 'DefineTiers' GROUP POOL 'TIERS' IS 'system' LIMIT(70)THEN 'gold' LIMIT(75)THEN 'silver'
RULE 'Rebalance' MIGRATE FROM POOL 'TIERS' TO POOL 'TIERS' WEIGHT(computeFileHeat
(CURRENT_TIMESTAMP-ACCESS_TIME,xattr('gpfs.FileHeat'), KB_ALLOCATED))FOR
FILESET('Object_Fileset')
WHERE NAME LIKE '%.data'
[I] 2015-11-22@02:30:20.045 Directory entries scanned: 1945.
[I] Directories scan: 1223 files, 594 directories, 128 other objects, 0 'skipped' files
and/or errors.
[I] 2015-11-22@02:30:20.050 Sorting 1945 file list records.
[I] Inodes scan: 1223 files, 594 directories, 128 other objects, 0 'skipped' files and/or
errors.
[I] 2015-11-22@02:30:20.345 Policy evaluation. 1945 files scanned.
[I] 2015-11-22@02:30:20.350 Sorting 1 candidate file list records.
[I] 2015-11-22@02:30:20.437 Choosing candidate files. 1 records scanned.
[I] Summary of Rule Applicability and File Choices:
  Rule#      Hit_Cnt      KB_Hit      Chosen      KB_Chosen      KB_I11      Rule
  0          98080572    3353660160    39384107    1328389120    292416      RULE
'Clean'
MIGRATE FROM POOL 'TIERS' WEIGHT(.) TO POOL 'TIERS' FOR FILESET(.) WHERE(.)

[I] Filesystem objects with no applicable rules: 1944.

[I] GPFS Policy Decisions and File Choice Totals:
  Chose to migrate 0KB: 1 of 1 candidates;
Predicted Data Pool Utilization in KB and %:
Pool_Name          KB_Occupied      KB_Total    Percent_Occupied
gold               169462784       6836715520   2.478716330%
silver              136192        13673431040   0.000996034%
system              8990720        13673431040   0.065753211%

```

6. Run the following command when you have no errors:

```
mmapplypolicy fs1 -P object_file_heat -I yes
```

For more information, see [“File heat: Tracking file access temperature” on page 579](#).

Chapter 34. Managing GPFS quotas

The GPFS quota system helps you to control the allocation of files and data blocks in a file system.

GPFS quotas can be defined for:

- Individual users
- Groups of users
- Individual filesets

Quotas are enabled by the system administrator when control over the amount of space used by the individual users, groups of users, or individual filesets is required. By default, user and group quota limits are enforced across the entire file system. Optionally, the scope of quota enforcement can be limited to an individual fileset boundary.

Quota accounting depends on a consistent mapping between user names and their numeric identifiers. This means that a single user who is accessing a quota-enabled file system from different nodes must be mapped to the same numeric user identifier from each node. Within a local cluster, the mapping is done by ensuring that /etc/passwd and /etc/group are identical across the cluster. When accessing quota-enabled file systems from other clusters, ensure that individual users who are accessing the quota-enabled file system have equivalent entries in /etc/passwd and /etc/group. Or, use the user identity mapping facility that is outlined in the [UID Mapping for GPFS in a Multi-cluster Environment IBM white paper](#) (https://www.ibm.com/docs/en/storage-scale?topic=STXKQY/uid_gpfs.pdf).

Note: A large number of quota records per file system can result from the following scenarios:

- There are a very large number of users, groups, or filesets.
- If the **--perfileset-quota** option is enabled, the number of possible quota records is the number of filesets times number of users (and groups).

GUI navigation

To work with this function in the GUI, log on to the IBM Storage Scale GUI and select **Files > Quotas**.

Quota related tasks include:

- [“Enabling and disabling GPFS quota management” on page 425](#)
- [“Default quotas” on page 427](#)
- [“Explicitly establishing and changing quotas” on page 431](#)
- [“Checking quotas” on page 436](#)
- [“Listing quotas” on page 438](#)
- [“Activating quota limit checking” on page 440](#)
- [“Deactivating quota limit checking” on page 441](#)
- [“Changing the scope of quota limit checking” on page 443](#)
- [“Creating file system quota reports” on page 444](#)
- [“Restoring quota files” on page 445](#)

For GPFS fileset quotas, see [“Filesets” on page 582](#).

Note: Windows nodes may be included in clusters that use GPFS quotas. However, Windows nodes do not support the quota commands.

Enabling and disabling GPFS quota management

You can enable GPFS quota management on new or existing GPFS file systems, establish quota values, and disable quota management by following the steps in this topic.

To enable GPFS quota management on a new GPFS file system:

- Specify the **-Q yes** option on the **mmcrfs** command. This option automatically activates quota enforcement whenever the file system is mounted. If you want the scope of quota limit enforcement to be based on individual filesets (rather than the entire file system), also specify the **--perfileset-quota** option on the **mmcrfs** command.
- Mount the file system.
- Issue the **mmedquota** or **mmsetquota** command to explicitly set quota values for users, groups, or filesets. See “[Explicitly establishing and changing quotas](#)” on page 431.

To enable GPFS quota management on an existing GPFS file system:

- Run the **mmchfs -Q yes** command. This command automatically activates quota enforcement whenever the file system is mounted or activates all subsequent mounts following the new quota setting if the file system is not mounted. If you want the scope of quota limit enforcement to be based on individual filesets (rather than the entire file system), also specify the **--perfileset-quota** option on the **mmchfs** command.

If an online **mmchfs -Q yes/no** command fails or is interrupted for any reason, **mmcheckquota** or **mmchfs -Q yes/no** must be rerun so that quota configuration for all nodes in the cluster will be brought into a consistent state.

All subsequent mounts will follow the new quota setting.

Note: The **perfileset-quota** cannot be enabled online in GPFS 4.1.

- Compile inode and disk block statistics using the **mmcheckquota** command. See “[Checking quotas](#)” on page 436. The values obtained can be used to establish realistic quota values when issuing the **mmedquota** or **mmsetquota** command.
- Issue the **mmedquota** or **mmsetquota** command to explicitly set quota values for users, groups, or filesets. See “[Explicitly establishing and changing quotas](#)” on page 431.

Once GPFS quota management has been enabled, you may establish quota values by:

- Setting default quotas for all new users, groups of users, or filesets.
- Explicitly establishing or changing quotas for users, groups of users, or filesets.
- Using the **gpfs_quotactl()** subroutine.

If **ignoreReplicationForQuota** is enabled, the quota commands ignore data replication factor.

To disable quota management, run the **mmchfs -Q no** command. All subsequent mounts will obey the new quota setting.

For complete usage information, see the *mmcheckquota command*, the *mmchfs command*, the *mmcrfs command*, and the *mmedquota command* in the *IBM Storage Scale: Command and Programming Reference Guide*. For additional information on quotas, see the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Related concepts

[Default quotas](#)

Default quota limits can be set for new users, groups, and filesets for a specified file system. Default quota limits can also be applied at a more granular level for new users and groups in a specified fileset.

[Implications of quotas for different protocols](#)

Quotas can mean different things for different protocols. This section describes how quotas affect the SMB and NFS protocols.

[Setting quotas for users on a per-project basis](#)

A file system must be properly configured in order to set quotas for users. Use this information to set quotas for any number of users on a per-project basis across protocols.

[Listing quotas](#)

The **mm1quota** command displays the file system quota limits, default quota limits, and current usage information.

Related tasks

[Explicitly establishing and changing quotas](#)

Use the **mmedquota** command to explicitly establish or change file system quota limits for users, groups of users, or filesets.

[Checking quotas](#)

The **mmcheckquota** command counts inode and space usage for a file system and writes the collected data into quota files.

[Activating quota limit checking](#)

Quota limit checking can be activated for users, groups, or fileset. Quota limit checking can also be activated for any combination of users, groups, and filesets.

[Deactivating quota limit checking](#)

During normal operation, there is no need to deactivate quota enforcement. The only reason that you might have to deactivate quota enforcement is when users are denied allocation that their quotas should allow, due to loss of quota information during node failure.

[Changing the scope of quota limit checking](#)

The scope of quota enforcement is established when quotas are activated. By default, user and group quota limits are enforced across the entire file system. Optionally, the scope of quota enforcement can be limited to an individual fileset boundary.

[Creating file system quota reports](#)

You can have GPFS prepare a quota report for a file system by using the **mmrepquota** command.

[Restoring quota files](#)

The method that is used for restoring GPFS quota files depends on the version of GPFS.

[Managing quota by using GUI](#)

You can create new quotas or modify existing ones. A quota is the amount of disk space and the amount of metadata that is assigned as upper limits for a specified user, group of users, or fileset. The IBM Storage Scale management GUI provides options to manage both capacity and inode quotas.

Default quotas

Default quota limits can be set for new users, groups, and filesets for a specified file system. Default quota limits can also be applied at a more granular level for new users and groups in a specified fileset.

When default quotas are managed at the fileset level, those quotas have a higher priority than those set at the file system level. If the status of the fileset-level defaults for one fileset is **Initial**, they will inherit default limits from global fileset-level defaults. The status of newly added fileset-level default quotas can be one of the following:

Initial

When the fileset is created, it will be in this state. All user and group quota accounts under the fileset will not follow the fileset defaults.

Quota on

All user and group quota accounts under the fileset that are created later will follow the fileset quota limits.

Quota off

All user and group quota accounts under the fileset that are created later will not follow the fileset quota limits. The users and groups will follow global fileset-level defaults if they are valid. If those defaults are not valid, the status will be initial.

Specific default quota recommendations for protocols:

- Since the protocols may have vastly different fileset requirements, it is not recommended to use default quotas at the fileset level. Rather, set explicit quotas and limits for each fileset in use by any and all protocols on a case-by-case basis.

- NFS: Prepare a default quota stanza file template, and at NFS export creation time, apply the default user or group quotas to the export path (assuming the export is an independent fileset) using per-fileset default quotas.
- SMB: Prepare a default quota stanza file template, and at SMB share creation time, apply the default user or group quotas to the share path (assuming the export is an independent fileset) using per-fileset default quotas.
- Object: IBM recommends using a single independent fileset, objectfs, for the object container. See *IBM Redpaper: A Deployment Guide for IBM Storage Scale Object* for details. With regard to quotas, here are the relevant sections from the Redpaper:
 - GPFS quotas: The amount of disk space and the number of inodes that are assigned as upper limits for a specified user, group of users, or fileset. With OpenStack Swift, GPFS user quotas are not used; instead, the system relies on OpenStack Swift quotas to provide a similar type of service. However, GPFS fileset quotas can still be defined (for example, for inodes, to limit the resources that are consumed by the fileset). See *Chapter 1.3 Key concepts and terminology* of the IBM Redpaper for details.
 - Swift quotas: Allows specification of the amount of disk space or number of objects that can be consumed by either an account (and subsequently all of its containers) or an individual container. The interaction between Swift quotas and GPFS quotas are described in more detail in *Chapter 6 Swift feature overview* and *Chapter 1.3 Key concepts and terminology* of the IBM Redpaper.
 - Quotas: Swift quotas allow a specific amount of disk capacity to be allocated to either containers or accounts by using Swift quotas. They also allow a limit on the maximum number of objects to be specified for containers or accounts. See *Chapter 6 Swift feature overview* of the IBM Redpaper for details.

Note: Although GPFS quotas do not explicitly interact with Swift quotas, it still might be useful to employ GPFS quotas to limit the amount of space or the number of inodes that is consumed by the object store. To do this, define GPFS quotas on the top-level independent fileset by specifying the maximum size or maximum inode usage that the object store can consume. See *Chapter 6 Swift feature overview* of the IBM Redpaper for details.

To enable default quota values:

1. Ensure the file system is configured correctly to use quotas:
 - a. The `-Q yes` option must be in effect for the file system.
 - b. To set default quotas at the fileset level, the `--perfileset-quota` option must also be in effect.

Note: If `--perfileset-quota` is in effect, all users and groups in the fileset `root` will not be impacted by default quota unless they are explicitly set.

The `-Q yes` and `--perfileset-quota` options are specified when creating a file system with the **mmcrfs** command or changing file system attributes with the **mmchfs** command. Use the **mmlsfs** command to display the current settings of these quota options.

2. Enable default quotas with the **mmdefquotaon** command.
3. Specify default quota values for new users, groups, and filesets by issuing the **mmdefedquota** command using a default quota stanza file. A single invocation of the **mmsetquota** command using a quota stanza file can perform the following operations:
 - Set default user quotas on a file system.
 - Set default group quotas on a file system.
 - Set a default perfileset user quota on a fileset (if `--perfileset-quota` is in effect).

The stanza file `/tmp/defaultQuotaExample` may look like this:

```
%quota:
  device=fs1
  command=setDefaultQuota
  type=USR
  blockQuota=25G
```

```
blockLimit=30G  
filesQuota=10K  
filesLimit=11K
```

```
%quota:  
device=fs1  
command=setDefaultQuota  
type=GRP  
blockQuota=75G  
blockLimit=90G  
filesQuota=30K  
filesLimit=33K
```

```
%quota:  
device=fs1  
command=setDefaultQuota  
type=USR  
fileset=fset0  
blockQuota=25G  
blockLimit=30G  
filesQuota=10K  
filesLimit=11K
```

Then issue the command:

```
# mmsetquota -F /tmp/defaultQuotaExample
```

4. To activate quota checking, use the **mmquotaon** command.
5. To list quotas, use the **mmlsquota** command.

The default quotas can be deactivated by issuing the **mmdefquotaoff** command.

For fileset recommendations, see “[Filesets and quotas](#)” on page 583.

For complete usage information, see the *mmchfs command*, *mmcrfs command*, *mmdefedquota command*, *mmdefquotaoff command*, *mmdefquotaon command*, *mmedquota command*, *mmlsfs command*, and *mmsetquota command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

[Implications of quotas for different protocols](#)

Quotas can mean different things for different protocols. This section describes how quotas affect the SMB and NFS protocols.

[Setting quotas for users on a per-project basis](#)

A file system must be properly configured in order to set quotas for users. Use this information to set quotas for any number of users on a per-project basis across protocols.

[Listing quotas](#)

The **mmlsquota** command displays the file system quota limits, default quota limits, and current usage information.

Related tasks

[Enabling and disabling GPFS quota management](#)

You can enable GPFS quota management on new or existing GPFS file systems, establish quota values, and disable quota management by following the steps in this topic.

[Explicitly establishing and changing quotas](#)

Use the **mmedquota** command to explicitly establish or change file system quota limits for users, groups of users, or filesets.

[Checking quotas](#)

The **mmcheckquota** command counts inode and space usage for a file system and writes the collected data into quota files.

[Activating quota limit checking](#)

Quota limit checking can be activated for users, groups, or fileset. Quota limit checking can also be activated for any combination of users, groups, and filesets.

Deactivating quota limit checking

During normal operation, there is no need to deactivate quota enforcement. The only reason that you might have to deactivate quota enforcement is when users are denied allocation that their quotas should allow, due to loss of quota information during node failure.

Changing the scope of quota limit checking

The scope of quota enforcement is established when quotas are activated. By default, user and group quota limits are enforced across the entire file system. Optionally, the scope of quota enforcement can be limited to an individual fileset boundary.

Creating file system quota reports

You can have GPFS prepare a quota report for a file system by using the `mmrepquota` command.

Restoring quota files

The method that is used for restoring GPFS quota files depends on the version of GPFS.

Managing quota by using GUI

You can create new quotas or modify existing ones. A quota is the amount of disk space and the amount of metadata that is assigned as upper limits for a specified user, group of users, or fileset. The IBM Storage Scale management GUI provides options to manage both capacity and inode quotas.

Implications of quotas for different protocols

Quotas can mean different things for different protocols. This section describes how quotas affect the SMB and NFS protocols.

Quotas are stored and enforced in the file system. See [Chapter 34, “Managing GPFS quotas,” on page 425](#) for details on how to enable and use quotas.

- SMB protocol and quotas

For SMB clients, quotas can limit the used and free space reported to clients:

- If the SMB option `gpfs : dfreequota` is set, the SMB server queries the user quota for the current user and the group quota to determine the available space. The group quota queried is for the primary group of the current user, unless the query is for a directory with the SGID ("set group id") bit set, in which case the owning group from the directory is used for the group quota query. The reason here is that with the SGID bit set on a directory, new files and subdirectories are created with the owning group from the directory, not the primary group of the user:
 - If the block limit is reached, the free space is reported as 0 and the size of the share is reported with the currently used data.
 - If the soft block quota is exceeded for longer than the block grace time, the free space is reported as 0 and the size of the share is reported with the currently used data.
 - If no limit is exceeded, the free space is reported as the free space according to the lowest quota limit.
 - If no quota is in place, the size and free space as queried from the underlying file system are reported.
 - In the case of per-fileset user and group quotas, the quotas are only queried from the root folder of the export. If a subdirectory inside the share is in a different fileset, the user and group quotas are not considered for the free space report.

Note: For including fileset quotas in the reported free space, configure the underlying file system with the `--filesetdf` flag (in `mmcdfs` or `mmchfs`). It is not possible to query or change individual quotas from a SMB client system.

- NFS protocol and quotas

It is not possible to query or change individual quotas from an NFS client system. User and group quotas are not included in the reported free space to a client. To include fileset quotas in the reported space to a client, configure the underlying file system with the **--filesetdf** flag (in **mmcrfs** or **mmchfs**).

Related concepts

[Default quotas](#)

Default quota limits can be set for new users, groups, and filesets for a specified file system. Default quota limits can also be applied at a more granular level for new users and groups in a specified fileset.

[Setting quotas for users on a per-project basis](#)

A file system must be properly configured in order to set quotas for users. Use this information to set quotas for any number of users on a per-project basis across protocols.

[Listing quotas](#)

The **mm1quota** command displays the file system quota limits, default quota limits, and current usage information.

Related tasks

[Enabling and disabling GPFS quota management](#)

You can enable GPFS quota management on new or existing GPFS file systems, establish quota values, and disable quota management by following the steps in this topic.

[Explicitly establishing and changing quotas](#)

Use the **mmedquota** command to explicitly establish or change file system quota limits for users, groups of users, or filesets.

[Checking quotas](#)

The **mmcheckquota** command counts inode and space usage for a file system and writes the collected data into quota files.

[Activating quota limit checking](#)

Quota limit checking can be activated for users, groups, or fileset. Quota limit checking can also be activated for any combination of users, groups, and filesets.

[Deactivating quota limit checking](#)

During normal operation, there is no need to deactivate quota enforcement. The only reason that you might have to deactivate quota enforcement is when users are denied allocation that their quotas should allow, due to loss of quota information during node failure.

[Changing the scope of quota limit checking](#)

The scope of quota enforcement is established when quotas are activated. By default, user and group quota limits are enforced across the entire file system. Optionally, the scope of quota enforcement can be limited to an individual fileset boundary.

[Creating file system quota reports](#)

You can have GPFS prepare a quota report for a file system by using the **mmrepquota** command.

[Restoring quota files](#)

The method that is used for restoring GPFS quota files depends on the version of GPFS.

[Managing quota by using GUI](#)

You can create new quotas or modify existing ones. A quota is the amount of disk space and the amount of metadata that is assigned as upper limits for a specified user, group of users, or fileset. The IBM Storage Scale management GUI provides options to manage both capacity and inode quotas.

Explicitly establishing and changing quotas

Use the **mmedquota** command to explicitly establish or change file system quota limits for users, groups of users, or filesets.

When setting quota limits for a file system, replication within the file system should be considered. See “[Listing quotas](#)” on page 438.

If `ignoreReplicationForQuota` is enabled, the quota commands ignore data replication factor.

The **mmedquota** command opens a session using your default editor, and prompts you for soft and hard limits for blocks and inodes. For example, to set user quotas for user **jesmith**, enter:

```
mmedquota -u jesmith
```

The system displays information in your default editor similar to:

```
*** Edit quota limits for USR jesmith:  
NOTE: block limits will be rounded up to the next multiple block size.  
      block units may be: K, M, G, T or P, inode units may be: K, M or G.  
gpfs0: blocks in use: 24576K, limits (soft = 0K, hard = 0K)  
      inodes in use: 0, limits (soft = 0K, hard = 0K)
```

Note: A quota limit of zero indicates that **no** quota limits are established.

The current (in use) block and inode usage is for display only, and it cannot be changed. When establishing a new quota, zeros appear as limits. Replace the zeros, or old values if you are changing existing limits, with values based on the user's needs and the resources available. When you close the editor, GPFS checks the values and applies them. If an invalid value is specified, GPFS generates an error message. If this occurs, reenter the **mmedquota** command. If the scope of quota limit enforcement is the entire file system, **mmedquota** lists all instances of the same user (for example, **jesmith**) on different GPFS file systems. If the quota enforcement is on a per-fileset basis, **mmedquota** lists all instances of the same user on different filesets on different GPFS file systems.

You might find it helpful to maintain a *quota prototype*, a set of limits that you can apply by name to any user, group, or fileset without entering the individual values manually. This makes it easy to set the same limits for all. The **mmedquota** command includes the **-p** option for naming a prototypical user, group, or fileset on which limits are to be based. The **-p** flag can be used only to propagate quotas from filesets within the same file system.

For example, to set group quotas for all users in a group that is named **blueteam** to the prototypical values established for **prototeam**, issue:

```
mmedquota -g -p prototeam blueteam
```

You can also reestablish default quotas for a specified user, group of users, or fileset when you use the **-d** option on the **mmedquota** command.

Note: You can use the **mmsetquota** command as an alternative to the **mmedquota** command.

For complete usage information, see the *mmedquota command* and the *mmsetquota command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

Default quotas

Default quota limits can be set for new users, groups, and filesets for a specified file system. Default quota limits can also be applied at a more granular level for new users and groups in a specified fileset.

Implications of quotas for different protocols

Quotas can mean different things for different protocols. This section describes how quotas affect the SMB and NFS protocols.

Setting quotas for users on a per-project basis

A file system must be properly configured in order to set quotas for users. Use this information to set quotas for any number of users on a per-project basis across protocols.

Listing quotas

The **mm1squota** command displays the file system quota limits, default quota limits, and current usage information.

Related tasks

Enabling and disabling GPFS quota management

You can enable GPFS quota management on new or existing GPFS file systems, establish quota values, and disable quota management by following the steps in this topic.

Checking quotas

The **mmcheckquota** command counts inode and space usage for a file system and writes the collected data into quota files.

Activating quota limit checking

Quota limit checking can be activated for users, groups, or fileset. Quota limit checking can also be activated for any combination of users, groups, and filesets.

Deactivating quota limit checking

During normal operation, there is no need to deactivate quota enforcement. The only reason that you might have to deactivate quota enforcement is when users are denied allocation that their quotas should allow, due to loss of quota information during node failure.

Changing the scope of quota limit checking

The scope of quota enforcement is established when quotas are activated. By default, user and group quota limits are enforced across the entire file system. Optionally, the scope of quota enforcement can be limited to an individual fileset boundary.

Creating file system quota reports

You can have GPFS prepare a quota report for a file system by using the **mmrepquota** command.

Restoring quota files

The method that is used for restoring GPFS quota files depends on the version of GPFS.

Managing quota by using GUI

You can create new quotas or modify existing ones. A quota is the amount of disk space and the amount of metadata that is assigned as upper limits for a specified user, group of users, or fileset. The IBM Storage Scale management GUI provides options to manage both capacity and inode quotas.

Setting quotas for users on a per-project basis

A file system must be properly configured in order to set quotas for users. Use this information to set quotas for any number of users on a per-project basis across protocols.

1. Ensure the file system is configured correctly to use quotas:

- a. The **-Q yes** option must be in effect for the file system.
- b. To set default quotas at the fileset level, the **--perfileset-quota** option must also be in effect.

Note: If **--perfileset-quota** is in effect, all users and groups in the fileset **root** are not impacted by default quota unless they are explicitly set.

The **-Q yes** and **--perfileset-quota** options are specified when creating a file system with the **mmcrfs** command or changing file system attributes with the **mmchfs** command. Use the **mmlsfs** command to display the current settings of these quota options.

Here are some examples:

- a. A GPFS cluster is created with configuration profile file, **example.profile**, which contains the following lines:

```
%filesystem
quotasAccountingEnabled=yes
quotasEnforced=user;group;fileset
perfilesetQuotas=yes
```

When a file system is created, those quota attributes are set automatically. Quota accounting is enabled on a perfileset basis for users and groups, and quotas are automatically enforced. This means that when a quota is reached, the end user will not be able to add more data to the file system.

mmcrfs fs5 nsd8

A listing of the file system config, by using the **mmlsfs** command, shows the following attributes and values, that are set by the **mmcrfs** command:

mmlsfs fs5

```
...  
-Q           user;group;fileset      Quotas accounting enabled  
          user;group;fileset      Quotas enforced  
          none                  Default quotas enabled  
--perfileset-quota Yes             Per-fileset quota enforcement  
....
```

For more information on **mmcrcluster** user-defined profiles, see *mmcrcluster command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

- b. Whether a GPFS cluster was created with a configuration profile file, a GPFS file system can be created with the quota attributes to be set. This can be done by calling the configuration profile file explicitly from the command line:

mmcrfs fs6 nsd9 --profile=example

For more information on user-defined profiles, see *mmcrfs command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

2. Create a fileset on the file system for the project by using the **mmcrfileset** command.

For example:

mmcrfileset fs5 projectX --inode-space=new

Note: It is recommended to create an independent fileset for the project.

3. Link the fileset by using the **mmlinkfileset** command.

The file system, **fs5**, must be mounted, by using the **mmmount** command. For example:

mmmount fs5 -a**mmchfs fs5 --inode-limit 400000:300000**

Output:

```
Set maxInodes for inode space 0 to 400000  
Fileset root changed.
```

mmlinkfileset fs5 projectX -J /gpfs/fs5/projectX

4. Create export/share by using the newly created fileset as the export/share path. For more information, see the *mmnfs command* and the *mmsmb command* in the *IBM Storage Scale: Command and Programming Reference Guide*. For example:

mmnfs export add /gpfs/fs5/projectX

5. If needed, specify absolute fileset inode limits by using the **mmchfileset** command. A fileset inode limit is analogous to saying this is how many files and directories the project is likely to produce. This is not something that can be easily recommended. Nonetheless, here is an example of how that can be set and listed for the file system and fileset:

mmchfileset fs5 projectX --inode-limit 200000

Output:

```
Set maxInodes for inode space 1 to 200000  
Fileset projectX changed.
```

mmlsfileset fs5 -L

Output:

Filesets in file system 'fs5':					Inode Space	MaxInodes	AllocInodes
Name	Id	RootInode	ParentId	Created			
Comment							

```

root      0   3      --      Sat Mar 28 13:40:33 2015  0   400000  310656    root
fileset
projectX 1   524291  --      Sat Mar 28 14:54:13 2015  1   200000  100032

```

6. Now that there is a fileset limit in place, which is entirely optional, to set group quota limits on the project, that is, on fileset **projectX** on file system **fs5**, use the **mmsetquota** command. For example, if the group **groupY** will access **projectX**:

mmsetquota fs5:projectX --group groupY --block 128G --files 150K

Here, the perfileset quota needs to be enabled on **fs5** as in step 1, and the group **groupY** must have a GID (group ID) on the GPFS cluster. The **block** parameter is used to specify the maximum size of the data on the storage device and the **files** parameter is used to specify the maximum number for files (or directories) the **groupY** is able to consume or create on **projectX**, a fileset of file system **fs5** that is exported through NFS in this example.

At this point, the quota accounting needs to be refreshed on the file system using the **mmcheckquota** command, and then a reporting of the quota limits on **projectX** can take place using the **mmrepquota** command. For example:

mmcheckquota fs5

mmrepquota fs5:projectX

Output:

Block Limits							File				
Limits	Name	fileset	type	KB	quota	limit	in_doubt	grace	files	quota	limit
in_doubt	root	projectX	USR	0	0	0	0	none	1	0	0
0	root	projectX	GRP	0	0	0	0	none	1	0	0
0	groupY	projectX	GRP	0	134217728	0	0	none	0	153600	0
0	none										

7. If the project grows, or shrinks, and quota changes at the group level are needed, the **mmsetquota** command can again be used to change the quotas for **groupY** on **projectX**. For example, if the expected limits for **projectX** doubles:

mmsetquota fs5:projectX --group groupY --block 256G --files 300K

mmrepquota fs5:projectX

Output:

Block Limits							File				
Limits	Name	fileset	type	KB	quota	limit	in_doubt	grace	files	quota	limit
in_doubt	root	projectX	USR	0	0	0	0	none	1	0	0
0	root	projectX	GRP	0	0	0	0	none	1	0	0
0	groupY	projectX	GRP	0	268435456	0	0	none	0	307200	0
0	none										

8. If the project is projected to exceed the inode limits for the fileset and file system, then these can also be adjusted upwards. For more information, see the *mmchfs command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

Default quotas

Default quota limits can be set for new users, groups, and filesets for a specified file system. Default quota limits can also be applied at a more granular level for new users and groups in a specified fileset.

Implications of quotas for different protocols

Quotas can mean different things for different protocols. This section describes how quotas affect the SMB and NFS protocols.

Listing quotas

The **mm1squota** command displays the file system quota limits, default quota limits, and current usage information.

Related tasks

Enabling and disabling GPFS quota management

You can enable GPFS quota management on new or existing GPFS file systems, establish quota values, and disable quota management by following the steps in this topic.

Explicitly establishing and changing quotas

Use the **mmcdquota** command to explicitly establish or change file system quota limits for users, groups of users, or filesets.

Checking quotas

The **mmccheckquota** command counts inode and space usage for a file system and writes the collected data into quota files.

Activating quota limit checking

Quota limit checking can be activated for users, groups, or fileset. Quota limit checking can also be activated for any combination of users, groups, and filesets.

Deactivating quota limit checking

During normal operation, there is no need to deactivate quota enforcement. The only reason that you might have to deactivate quota enforcement is when users are denied allocation that their quotas should allow, due to loss of quota information during node failure.

Changing the scope of quota limit checking

The scope of quota enforcement is established when quotas are activated. By default, user and group quota limits are enforced across the entire file system. Optionally, the scope of quota enforcement can be limited to an individual fileset boundary.

Creating file system quota reports

You can have GPFS prepare a quota report for a file system by using the **mmrepquota** command.

Restoring quota files

The method that is used for restoring GPFS quota files depends on the version of GPFS.

Managing quota by using GUI

You can create new quotas or modify existing ones. A quota is the amount of disk space and the amount of metadata that is assigned as upper limits for a specified user, group of users, or fileset. The IBM Storage Scale management GUI provides options to manage both capacity and inode quotas.

Checking quotas

The **mmccheckquota** command counts inode and space usage for a file system and writes the collected data into quota files.

You must use the **mmccheckquota** command if any of the following are true:

1. Quota information is lost due to node failure.

Node failure might leave users unable to open files or deny them disk space that their quotas should allow.

2. The *in doubt* value approaches the quota limit. To see the *in doubt* value, use the **mm1squota** or **mmrepquota** commands.

As the sum of the *in doubt* value and the current usage cannot exceed the hard limit, the actual block space and number of files available to the user, group, or fileset might be constrained by the *in doubt* value. Should the *in doubt* value approach a significant percentage of the quota, use the **mmccheckquota** command to account for the lost space and files.

Note: Running **mmcheckquota** is also recommended (in an appropriate time slot) if the following message is displayed after running **mmrepquota**, **mmlsquota**, or **mmedquota**:

```
Quota accounting information is inaccurate and quotacheck must be run.
```

When issuing the **mmcheckquota** command on a mounted file system, negative *in doubt* values might be reported if the quota server processes a combination of up-to-date and back-level information. This is a transient situation and can be ignored.

During the normal operation of file systems with quotas enabled (not running **mmcheckquota** online), the usage data reflects the actual usage of the blocks and inodes in the sense that if you delete files you should see the usage amount decrease. The *in doubt* value does not reflect how much the user has used already, it is just the number of quotas that the quota server has assigned to its clients. The quota server does not know whether the assigned amount has been used or not. The only situation where the *in doubt* value is important to the user is when the sum of the usage and the *in doubt* value is greater than the user's quota hard limit. In this case, the user is not allowed to allocate more blocks or inodes unless they bring the usage down.

For example, to check quotas for the file system **fs1** and report differences between calculated and recorded disk quotas, enter:

```
mmcheckquota -v fs1
```

The information displayed shows that the quota information for **USR7** was corrected. Due to a system failure, this information was lost at the server, which recorded 0 subblocks and 0 files. The current usage data that is counted is 96 subblocks and 3 files. This is used to update the quota:

```
fs1: quota check found the following differences:  
USR7: 96 subblocks counted (was 0); 3 inodes counted (was 0)
```

Note: In cases where small files do not have an extra block that is allocated for them, quota usage might show less space usage than expected.

For complete usage information, see the *mmcheckquota command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

Default quotas

Default quota limits can be set for new users, groups, and filesets for a specified file system. Default quota limits can also be applied at a more granular level for new users and groups in a specified fileset.

Implications of quotas for different protocols

Quotas can mean different things for different protocols. This section describes how quotas affect the SMB and NFS protocols.

Setting quotas for users on a per-project basis

A file system must be properly configured in order to set quotas for users. Use this information to set quotas for any number of users on a per-project basis across protocols.

Listing quotas

The **mmlsquota** command displays the file system quota limits, default quota limits, and current usage information.

Related tasks

Enabling and disabling GPFS quota management

You can enable GPFS quota management on new or existing GPFS file systems, establish quota values, and disable quota management by following the steps in this topic.

Explicitly establishing and changing quotas

Use the **mmedquota** command to explicitly establish or change file system quota limits for users, groups of users, or filesets.

Activating quota limit checking

Quota limit checking can be activated for users, groups, or fileset. Quota limit checking can also be activated for any combination of users, groups, and filesets.

Deactivating quota limit checking

During normal operation, there is no need to deactivate quota enforcement. The only reason that you might have to deactivate quota enforcement is when users are denied allocation that their quotas should allow, due to loss of quota information during node failure.

Changing the scope of quota limit checking

The scope of quota enforcement is established when quotas are activated. By default, user and group quota limits are enforced across the entire file system. Optionally, the scope of quota enforcement can be limited to an individual fileset boundary.

Creating file system quota reports

You can have GPFS prepare a quota report for a file system by using the **mmrepquota** command.

Restoring quota files

The method that is used for restoring GPFS quota files depends on the version of GPFS.

Managing quota by using GUI

You can create new quotas or modify existing ones. A quota is the amount of disk space and the amount of metadata that is assigned as upper limits for a specified user, group of users, or fileset. The IBM Storage Scale management GUI provides options to manage both capacity and inode quotas.

Listing quotas

The **mmlsquota** command displays the file system quota limits, default quota limits, and current usage information.

If the scope of quota limit enforcement is the entire file system, **mmlsquota -u** or **mmlsquota -g** lists all instances of the same user or group on different GPFS file systems. If the quota enforcement is on a per-fileset basis, **mmlsquota -u** or **mmlsquota -g** lists all instances of the same user or group on different filesets on different GPFS file systems.

If data replication is enabled in IBM Storage Scale, quota management includes the size of the replicated data in calculating the amount of data that is in use by the user, group, or fileset. For example, if the data replication factor is two and a user has used approximately 194 KiB of data, quota management calculates the amount of data in use as $2 * 194 \text{ KiB} = 388 \text{ KiB}$. (The 194 KiB is an approximation because quota management rounds up to the next subblock unit size.) Thus the amount of data in use that is reported by **mmlsquota** and **mmrepquota** (388 KiB) would be roughly twice the amount of data in use than is reported by commands such as **ls** and **du** (194 KiB).

Quota management does not include replicated data or metadata in calculating the number of files in use. If data replication is enabled, quota management replicates the file data and includes the replicated size in calculating the amount of data in use, as described in the previous paragraph. If `ignoreReplicationForQuota` is enabled, the quota commands ignore data replication factor. If metadata replication is enabled, quota management replicates the inode of each file but the number of files in use does not change. For example, if the data and metadata replication factors are both two and the user has created 92 files, the total number of files in use is still 92. (The number of inodes in use is also 92, because there is one inode per file. Copies of the 92 inodes are included in the replicated metadata.) Thus the number of files in use that is reported by **mmlsquota** and **mmrepquota** is the same regardless of whether data replication or metadata replication is enabled.

The **mmlsquota** command might report negative in-doubt values if quota management processes a combination of up-to-date and out-of-date information. This condition is transient and can be ignored.

To display the quota information for one user, one group of users, or one fileset, issue the **mmlsquota** command with the **-g**, **-u**, or **-j** option. If none of these options is specified, the command displays user quotas for the user who issues the command.

To display default quota information, issue the **mmlsquota** command with the **-d** option. For example, to display default quota information for users of all the file systems in the cluster, issue the following command:

```
mmlsquota -d -u
```

The following output is an example:

Default Block Limits(KB)			Default File Limits			Remarks
Filesystem	type	quota	limit	quota	limit	
fs1	USR	5242880	6291456	0	0	
fs2	USR	no default limits				

In this example, file system **fs1** shows that the default block quota for users is set at 5 GB for the soft limit and 6 GB for the hard limit. For file system **fs2**, no default quotas for users are set.

When **mmlsquota -d** is specified in combination with the **-u**, **-g**, or **-j** options, default file system quotas are displayed. When **mmlsquota -d** is specified without any of the **-u**, **-g**, or **-j** options, default fileset-level quotas are displayed.

If you issue the **mmlsquota** command with the **-e** option, quota management collects updated information from all the cluster nodes before it displays the output. If a node to which in-doubt space is allocated fails before it can update quota management with its actual amount of space, this space might not be included in the output. If the amount of in-doubt space approaches a significant percentage of the quota, issue the **mmcheckquota** command to account for the lost space.

To collect and display updated quota information about a group **blueteam**, issue the **mmlsquota** command with the **-g** and **-e** options:

```
mmlsquota -g blueteam -e
```

The following output is an example:

Block Limits				File Limits						
Filesystem	type	KB	quota	limit	in_doubt	grace	files	quota	limit	in_doubt
grace	Disk quotas for group blueteam (gid 100):									
fs1	GRP	45730	52000	99000	1335	none	411	580	990	19
none										

For complete usage information, see the *mmlsquota command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

Default quotas

Default quota limits can be set for new users, groups, and filesets for a specified file system. Default quota limits can also be applied at a more granular level for new users and groups in a specified fileset.

Implications of quotas for different protocols

Quotas can mean different things for different protocols. This section describes how quotas affect the SMB and NFS protocols.

Setting quotas for users on a per-project basis

A file system must be properly configured in order to set quotas for users. Use this information to set quotas for any number of users on a per-project basis across protocols.

Related tasks

Enabling and disabling GPFS quota management

You can enable GPFS quota management on new or existing GPFS file systems, establish quota values, and disable quota management by following the steps in this topic.

Explicitly establishing and changing quotas

Use the **mmcdquota** command to explicitly establish or change file system quota limits for users, groups of users, or filesets.

Checking quotas

The **mmcheckquota** command counts inode and space usage for a file system and writes the collected data into quota files.

Activating quota limit checking

Quota limit checking can be activated for users, groups, or fileset. Quota limit checking can also be activated for any combination of users, groups, and filesets.

Deactivating quota limit checking

During normal operation, there is no need to deactivate quota enforcement. The only reason that you might have to deactivate quota enforcement is when users are denied allocation that their quotas should allow, due to loss of quota information during node failure.

Changing the scope of quota limit checking

The scope of quota enforcement is established when quotas are activated. By default, user and group quota limits are enforced across the entire file system. Optionally, the scope of quota enforcement can be limited to an individual fileset boundary.

Creating file system quota reports

You can have GPFS prepare a quota report for a file system by using the **mmrepquota** command.

Restoring quota files

The method that is used for restoring GPFS quota files depends on the version of GPFS.

Managing quota by using GUI

You can create new quotas or modify existing ones. A quota is the amount of disk space and the amount of metadata that is assigned as upper limits for a specified user, group of users, or fileset. The IBM Storage Scale management GUI provides options to manage both capacity and inode quotas.

Activating quota limit checking

Quota limit checking can be activated for users, groups, or fileset. Quota limit checking can also be activated for any combination of users, groups, and filesets.

You can have quotas that are activated automatically whenever the file system is mounted by specifying the quota option (-Q yes) when creating (**mmcrlfs -Q yes**) or changing (**mmchfs -Q yes**) a GPFS file system. When creating a file system, the default is to **not** have quotas activated, so you must specify this option if you want quotas activated.

The **mmquotaon** command is used to turn quota limit checking back on if it is deactivated by issuing the **mmquotaooff** command. Specify the file system name, and whether user, group, or fileset quotas are to be activated. If you want all three fileset quotas activated (user, group, and fileset), specify only the file system name. After quotas are turned back on, issue the **mmcheckquota** command to count inode and space usage.

For example, to activate user quotas on the file system **fs1**, enter:

```
mmquotaon -u fs1
```

To confirm the change, enter:

```
mmlsfs fs1 -Q
```

The system displays output similar to:

flag	value	description
-Q	user	Quotas enforced

For complete usage information, see the *mmquotaon command* and the *mmlsfs command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

Default quotas

Default quota limits can be set for new users, groups, and filesets for a specified file system. Default quota limits can also be applied at a more granular level for new users and groups in a specified fileset.

Implications of quotas for different protocols

Quotas can mean different things for different protocols. This section describes how quotas affect the SMB and NFS protocols.

Setting quotas for users on a per-project basis

A file system must be properly configured in order to set quotas for users. Use this information to set quotas for any number of users on a per-project basis across protocols.

Listing quotas

The **mmquota** command displays the file system quota limits, default quota limits, and current usage information.

Related tasks

Enabling and disabling GPFS quota management

You can enable GPFS quota management on new or existing GPFS file systems, establish quota values, and disable quota management by following the steps in this topic.

Explicitly establishing and changing quotas

Use the **mmquotad** command to explicitly establish or change file system quota limits for users, groups of users, or filesets.

Checking quotas

The **mmcheckquota** command counts inode and space usage for a file system and writes the collected data into quota files.

Deactivating quota limit checking

During normal operation, there is no need to deactivate quota enforcement. The only reason that you might have to deactivate quota enforcement is when users are denied allocation that their quotas should allow, due to loss of quota information during node failure.

Changing the scope of quota limit checking

The scope of quota enforcement is established when quotas are activated. By default, user and group quota limits are enforced across the entire file system. Optionally, the scope of quota enforcement can be limited to an individual fileset boundary.

Creating file system quota reports

You can have GPFS prepare a quota report for a file system by using the **mmrepquota** command.

Restoring quota files

The method that is used for restoring GPFS quota files depends on the version of GPFS.

Managing quota by using GUI

You can create new quotas or modify existing ones. A quota is the amount of disk space and the amount of metadata that is assigned as upper limits for a specified user, group of users, or fileset. The IBM Storage Scale management GUI provides options to manage both capacity and inode quotas.

Deactivating quota limit checking

During normal operation, there is no need to deactivate quota enforcement. The only reason that you might have to deactivate quota enforcement is when users are denied allocation that their quotas should allow, due to loss of quota information during node failure.

If this occurs, use the **mmcheckquota** command after reactivating quotas to reconcile allocation data.

When quota enforcement is deactivated, disk space and file allocations are made without regard to limits.

The **mmquotaooff** command is used to deactivate quota limit checking. Specify the file system name and whether user, group, or fileset quotas, or any combination of these three, are to be deactivated. If you want all types of quotas that are deactivated, specify only the file system name.

For example, to deactivate only user quotas on the file system **fs1**, enter:

```
mmquotaoff -u fs1
```

To confirm the change, enter:

```
mmlsfs fs1 -Q
```

The system displays output similar to:

flag	value	description
-Q	group;fileset	Quotas enforced

For complete usage information, see the *mmquotaoff command* and the *mmlsfs command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

Default quotas

Default quota limits can be set for new users, groups, and filesets for a specified file system. Default quota limits can also be applied at a more granular level for new users and groups in a specified fileset.

Implications of quotas for different protocols

Quotas can mean different things for different protocols. This section describes how quotas affect the SMB and NFS protocols.

Setting quotas for users on a per-project basis

A file system must be properly configured in order to set quotas for users. Use this information to set quotas for any number of users on a per-project basis across protocols.

Listing quotas

The **mmlsquota** command displays the file system quota limits, default quota limits, and current usage information.

Related tasks

Enabling and disabling GPFS quota management

You can enable GPFS quota management on new or existing GPFS file systems, establish quota values, and disable quota management by following the steps in this topic.

Explicitly establishing and changing quotas

Use the **mmedquota** command to explicitly establish or change file system quota limits for users, groups of users, or filesets.

Checking quotas

The **mmcheckquota** command counts inode and space usage for a file system and writes the collected data into quota files.

Activating quota limit checking

Quota limit checking can be activated for users, groups, or fileset. Quota limit checking can also be activated for any combination of users, groups, and filesets.

Changing the scope of quota limit checking

The scope of quota enforcement is established when quotas are activated. By default, user and group quota limits are enforced across the entire file system. Optionally, the scope of quota enforcement can be limited to an individual fileset boundary.

Creating file system quota reports

You can have GPFS prepare a quota report for a file system by using the **mmrepquota** command.

Restoring quota files

The method that is used for restoring GPFS quota files depends on the version of GPFS.

Managing quota by using GUI

You can create new quotas or modify existing ones. A quota is the amount of disk space and the amount of metadata that is assigned as upper limits for a specified user, group of users, or fileset. The IBM Storage Scale management GUI provides options to manage both capacity and inode quotas.

Changing the scope of quota limit checking

The scope of quota enforcement is established when quotas are activated. By default, user and group quota limits are enforced across the entire file system. Optionally, the scope of quota enforcement can be limited to an individual fileset boundary.

The scope of quota enforcement can be changed using the `mmchfs` command and specifying either the `--perfileset-quota` or `--noperfileset-quota` option as needed.

After changing the scope of quota enforcement, `mmccheckquota` must be run to properly update the quota usage information.

Related concepts

Default quotas

Default quota limits can be set for new users, groups, and filesets for a specified file system. Default quota limits can also be applied at a more granular level for new users and groups in a specified fileset.

Implications of quotas for different protocols

Quotas can mean different things for different protocols. This section describes how quotas affect the SMB and NFS protocols.

Setting quotas for users on a per-project basis

A file system must be properly configured in order to set quotas for users. Use this information to set quotas for any number of users on a per-project basis across protocols.

Listing quotas

The `mmlsquota` command displays the file system quota limits, default quota limits, and current usage information.

Related tasks

Enabling and disabling GPFS quota management

You can enable GPFS quota management on new or existing GPFS file systems, establish quota values, and disable quota management by following the steps in this topic.

Explicitly establishing and changing quotas

Use the `mmcdquota` command to explicitly establish or change file system quota limits for users, groups of users, or filesets.

Checking quotas

The `mmccheckquota` command counts inode and space usage for a file system and writes the collected data into quota files.

Activating quota limit checking

Quota limit checking can be activated for users, groups, or fileset. Quota limit checking can also be activated for any combination of users, groups, and filesets.

Deactivating quota limit checking

During normal operation, there is no need to deactivate quota enforcement. The only reason that you might have to deactivate quota enforcement is when users are denied allocation that their quotas should allow, due to loss of quota information during node failure.

Creating file system quota reports

You can have GPFS prepare a quota report for a file system by using the `mmrepquota` command.

Restoring quota files

The method that is used for restoring GPFS quota files depends on the version of GPFS.

Managing quota by using GUI

You can create new quotas or modify existing ones. A quota is the amount of disk space and the amount of metadata that is assigned as upper limits for a specified user, group of users, or fileset. The IBM Storage Scale management GUI provides options to manage both capacity and inode quotas.

Creating file system quota reports

You can have GPFS prepare a quota report for a file system by using the `mmrepquota` command.

The quota report lists:

1. Number of files used
2. Amount of disk space used
3. Current quota limits
4. In doubt quotas (disk space that is allocated but currently unaccounted for)
5. Grace period allowance to exceed the soft limit
6. Whether the quotas are explicitly set (e), are default values at the file system level (d_fsys), are default values at the fileset level (d_fset), or initial values (i)

The entry type also indicates whether default quotas are enabled for the file system (default on or default off).

Specify whether you want to list only user quota information (-u flag), group quota information (-g flag), or fileset quota information (-j flag) in the `mmrepquota` command. The default is to summarize all three quotas. If the -e flag is not specified, there is the potential to display negative usage values because the quota server might process a combination of up-to-date and back-level information. See [“Listing quotas” on page 438](#).

If the scope of quota limit enforcement is the entire file system, `mmrepquota -u` or `mmrepquota -g` lists all users or groups on different GPFS file systems. If the quota enforcement is on a per-fileset basis, `mmrepquota -u` or `mmrepquota -g` lists all instances of the same user or group on different filesets on different GPFS file systems.

To list the group quotas (-g option) for all file systems in the cluster (-a option), and print a report with header lines (-v option), enter:

```
mmrepquota -g -v -a
```

The system displays information similar to:

```
*** Report for GRP quotas on fs1
      Block Limits                               File Limits
Name   type    KB  quota  limit  in_doubt grace |   files  quota  limit  in_doubt  grace
entryType
system  GRP    25088     0     0   209120  none |     32     0     0     1078  none
default  on
usr     GRP    435256     0     0   199712  none |     11     0     0      899  none
d_fsys
```

For complete usage information, see the `mmrepquota` command in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

Default quotas

Default quota limits can be set for new users, groups, and filesets for a specified file system. Default quota limits can also be applied at a more granular level for new users and groups in a specified fileset.

Implications of quotas for different protocols

Quotas can mean different things for different protocols. This section describes how quotas affect the SMB and NFS protocols.

Setting quotas for users on a per-project basis

A file system must be properly configured in order to set quotas for users. Use this information to set quotas for any number of users on a per-project basis across protocols.

Listing quotas

The **mm1squota** command displays the file system quota limits, default quota limits, and current usage information.

Related tasks

Enabling and disabling GPFS quota management

You can enable GPFS quota management on new or existing GPFS file systems, establish quota values, and disable quota management by following the steps in this topic.

Explicitly establishing and changing quotas

Use the **mmcdquota** command to explicitly establish or change file system quota limits for users, groups of users, or filesets.

Checking quotas

The **mmcheckquota** command counts inode and space usage for a file system and writes the collected data into quota files.

Activating quota limit checking

Quota limit checking can be activated for users, groups, or fileset. Quota limit checking can also be activated for any combination of users, groups, and filesets.

Deactivating quota limit checking

During normal operation, there is no need to deactivate quota enforcement. The only reason that you might have to deactivate quota enforcement is when users are denied allocation that their quotas should allow, due to loss of quota information during node failure.

Changing the scope of quota limit checking

The scope of quota enforcement is established when quotas are activated. By default, user and group quota limits are enforced across the entire file system. Optionally, the scope of quota enforcement can be limited to an individual fileset boundary.

Restoring quota files

The method that is used for restoring GPFS quota files depends on the version of GPFS.

Managing quota by using GUI

You can create new quotas or modify existing ones. A quota is the amount of disk space and the amount of metadata that is assigned as upper limits for a specified user, group of users, or fileset. The IBM Storage Scale management GUI provides options to manage both capacity and inode quotas.

Restoring quota files

The method that is used for restoring GPFS quota files depends on the version of GPFS.

The three scenarios for restoring GPFS quota files are as follows:

1. The file system version is lower than 4.1.0.0.

In scenario 1, quota files can be backed up directly by copying visible quota files and then restored using the **mmcheckquota** command. The newly-specified backup quota files are transferred from normal files to quota files (metadata). Old quota files are converted from metadata to "normal" files, so these old quota files can be deleted.

2. The file system version is 4.1.0.0 or higher, but lower than 4.1.1.0.

In scenario 2, quota files cannot be restored using the **mmcheckquota** command.

3. The file system version is 4.1.1.0 (or higher).

In scenario 3, quota files can be restored using the **mmcheckquota** command. Use the **mmcheckquota --backup** command to back up quota files. You can restore quota files from the former backup quota files. The **mmcheckquota** command copies data from specified backup quota files to "invisible" quota files. You cannot view or delete the original quota files. You can delete specified backup quota files only.

Additional details about the three scenarios for restoring GPFS quota files follow.

In scenarios 1 and 3:

- User, group, and fileset quota files can be restored from a backup copy of the original quota file. When restoring quota files, the backup file must be in the root directory of the GPFS file system.

In scenario 1, if a backup copy of the original quota file does not exist, an empty file will be created when the `mmcheckquota` command is issued.

In scenario 3, the `mmcheckquota` command does nothing and prints an error.

- The user, group, or fileset files can be restored from backup copies by issuing the `mmcheckquota` command with the appropriate options.

1. To restore the user quota file for the file system **fs1** from the backup file **userQuotaInfo**, enter:

```
mmcheckquota -u userQuotaInfo fs1
```

This command must be run offline (that is, no nodes are mounted).

2. This will restore the user quota limits set for the file system, but the usage information will not be current. To bring the usage information to current values, the command must be reissued:

```
mmcheckquota fs1
```

In scenario 1, if you want to nullify all quota configuration and then reinitialize it, follow these steps:

1. Remove the existing quota files that are corrupted.

a. Disable quota management:

```
mmchfs fs1 -Q no
```

b. Remove the `user.quota`, `group.quota`, and `fileset.quota` files.

2. Enable quota management.

a. Issue the following command:

```
mmchfs fs1 -Q yes
```

3. Reestablish quota limits by issuing the `mmedquota` command or the `mmdefedquota` command.

4. Gather the current quota usage values by issuing the `mmcheckquota` command.

In scenario 2, quota files do not exist externally. Therefore, use `mmbackupconfig` and `mmrestoreconfig` to restore quota configurations.

For complete usage information, see the *mmcheckquota command*, the *mmdefedquota command*, and the *mmedquota command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

[Default quotas](#)

Default quota limits can be set for new users, groups, and filesets for a specified file system. Default quota limits can also be applied at a more granular level for new users and groups in a specified fileset.

[Implications of quotas for different protocols](#)

Quotas can mean different things for different protocols. This section describes how quotas affect the SMB and NFS protocols.

[Setting quotas for users on a per-project basis](#)

A file system must be properly configured in order to set quotas for users. Use this information to set quotas for any number of users on a per-project basis across protocols.

[Listing quotas](#)

The **mm1quota** command displays the file system quota limits, default quota limits, and current usage information.

Related tasks

[Enabling and disabling GPFS quota management](#)

You can enable GPFS quota management on new or existing GPFS file systems, establish quota values, and disable quota management by following the steps in this topic.

[Explicitly establishing and changing quotas](#)

Use the **mmedquota** command to explicitly establish or change file system quota limits for users, groups of users, or filesets.

[Checking quotas](#)

The **mmcheckquota** command counts inode and space usage for a file system and writes the collected data into quota files.

[Activating quota limit checking](#)

Quota limit checking can be activated for users, groups, or fileset. Quota limit checking can also be activated for any combination of users, groups, and filesets.

[Deactivating quota limit checking](#)

During normal operation, there is no need to deactivate quota enforcement. The only reason that you might have to deactivate quota enforcement is when users are denied allocation that their quotas should allow, due to loss of quota information during node failure.

[Changing the scope of quota limit checking](#)

The scope of quota enforcement is established when quotas are activated. By default, user and group quota limits are enforced across the entire file system. Optionally, the scope of quota enforcement can be limited to an individual fileset boundary.

[Creating file system quota reports](#)

You can have GPFS prepare a quota report for a file system by using the **mmrepquota** command.

[Managing quota by using GUI](#)

You can create new quotas or modify existing ones. A quota is the amount of disk space and the amount of metadata that is assigned as upper limits for a specified user, group of users, or fileset. The IBM Storage Scale management GUI provides options to manage both capacity and inode quotas.

Managing quota by using GUI

You can create new quotas or modify existing ones. A quota is the amount of disk space and the amount of metadata that is assigned as upper limits for a specified user, group of users, or fileset. The IBM Storage Scale management GUI provides options to manage both capacity and inode quotas.

To define user, group, or fileset quota, you need to enable quota for the file system if it is already not enabled. You can enable quota for the file system from the **Enable Quota** option that is available in the Settings tab of the **Files > Quotas** page of the IBM Storage Scale GUI.

You can define user, group, and fileset quota from the **Files > Quotas** page. The following procedure explains how to define user quota:

1. Go to **Files > Quotas** page in the IBM Storage Scale GUI.

The following options are available under the various tabs of the Quotas page:

User

- Define soft and hard limit for the capacity and inode.
- Monitor quota.
- Create, modify, and delete quota.
- Change explicit quota to default quota.

Group

- Create group quota at the file system level. You can define a soft and hard limit for the capacity and inodes.
- Monitor group quota.
- Modify and delete group quota.
- Change the group quota to default group quota.

Fileset

- Create fileset quota at the file system level. You can define a soft and hard limit for the capacity and inodes.
- Monitor fileset quota.
- Modify and delete fileset quota.
- Change the fileset quota to default fileset quota.

Settings

- Enable or disable quota. You need to enforce quota at the file system level to enable user, group, and fileset quotas.
- Change the quota accounting scope. The quota accounting can be done either at the file system level or per fileset level. Users and groups can have a different quota limits in each fileset, if the accounting scope is *per fileset*.
- Check quota. This task takes significant amount of time to complete the process and it has some performance impact on the system.
- Change grace periods set for soft limit. You can define soft limit for the capacity and number of inodes. When the soft limit is reached, the system starts a grace period. Users cannot write any more data when the grace period is over, or the hard limit is reached.

Default Quota

- Modify the default quota set for user, group, and filesets.
- View the existing default quota definitions that are defined for each file system.

This procedure explains how to create user quota. So, the corresponding options are selected in the following steps.

2. From the **User** tab, click **Create User Quota**.
3. In the **File system** field, select the file system for which you need to define user quota.
4. Specify the user ID in the **User ID** field. The **Used Capacity** field displays the already used capacity by the user.
5. Define the soft limit in the **Soft Limit (Capacity)** field. The value must be between 256 KiB and the hard limit. At the soft quota limit, a grace period starts. Data can be written until the grace period is expired or the hard quota limit is reached.
6. Define the hard limit in the **Hard Limit (Capacity)** field. The hard capacity limit defines the maximum capacity that can be allocated for the fileset, user, or group. You cannot write more data when the quota limit is reached, unless you remove the existing files.
7. Specify the soft limit for inodes in the **Soft Limits (Inodes)** field.
8. Specify the hard limit for inodes in the **Hard Limit (Inodes)** field.
9. Click **Create**. The user quota is created and you can see it from the user quota table.

Related concepts

Default quotas

Default quota limits can be set for new users, groups, and filesets for a specified file system. Default quota limits can also be applied at a more granular level for new users and groups in a specified fileset.

Implications of quotas for different protocols

Quotas can mean different things for different protocols. This section describes how quotas affect the SMB and NFS protocols.

Setting quotas for users on a per-project basis

A file system must be properly configured in order to set quotas for users. Use this information to set quotas for any number of users on a per-project basis across protocols.

Listing quotas

The **mm1quota** command displays the file system quota limits, default quota limits, and current usage information.

Related tasks

Enabling and disabling GPFS quota management

You can enable GPFS quota management on new or existing GPFS file systems, establish quota values, and disable quota management by following the steps in this topic.

Explicitly establishing and changing quotas

Use the **mmcdquota** command to explicitly establish or change file system quota limits for users, groups of users, or filesets.

Checking quotas

The **mmccheckquota** command counts inode and space usage for a file system and writes the collected data into quota files.

Activating quota limit checking

Quota limit checking can be activated for users, groups, or fileset. Quota limit checking can also be activated for any combination of users, groups, and filesets.

Deactivating quota limit checking

During normal operation, there is no need to deactivate quota enforcement. The only reason that you might have to deactivate quota enforcement is when users are denied allocation that their quotas should allow, due to loss of quota information during node failure.

Changing the scope of quota limit checking

The scope of quota enforcement is established when quotas are activated. By default, user and group quota limits are enforced across the entire file system. Optionally, the scope of quota enforcement can be limited to an individual fileset boundary.

Creating file system quota reports

You can have GPFS prepare a quota report for a file system by using the **mmrepquota** command.

Restoring quota files

The method that is used for restoring GPFS quota files depends on the version of GPFS.

Chapter 35. Managing GUI users

GUI users of the IBM Storage Scale system can monitor, configure, and manage the IBM Storage Scale system. You can create users who can perform different administrative tasks on the system. Use the **Services > GUI > Users** page to create users.

Note: Only users with *SecurityAdmin* or *UserAdmin* role can create a GUI user.

User roles and permissions

Each GUI user must be part of a user group or multiple groups that are defined on the system. When you create a new user, you need to assign the user to one of the default user groups or to a custom user group. User groups are assigned with predefined roles that authorize the users within that group to a specific set of operations on the GUI.

Predefined roles are assigned to user groups to define the working scope within the GUI. If a user is assigned to more than one user group, the permissions are additive, not restrictive. The predefined role names cannot be changed. The following are the default user groups:

- **Administrator**

Manages all functions on the system except those deals with managing users, user groups, and authentication.

- **SecurityAdmin**

Manages all functions on the system, including managing users, user groups, and user authentication.

- **SystemAdmin**

Manages clusters, nodes, alert logs, and authentication.

- **StorageAdmin**

Manages disks, file systems, pools, filesets, and ILM policies.

- **SnapAdmin**

Manages snapshots for file systems and filesets.

- **DataAccess**

Controls access to data. For example, managing access control lists.

- **Monitor**

Monitors objects and system configuration but cannot configure, modify, or manage the system or its resources.

- **ProtocolAdmin**

Manages Object Storage and data export definitions of SMB and NFS protocols.

- **UserAdmin**

Manages access for GUI users. Users who are part of this group have edit permissions only in the Users, Groups and Password Policy tabs of the **Services > GUI** page of the GUI.

If a GUI node fails, the application fails over to the new node. The GUI master node fails over automatically.

- **CsiAdmin**

Manages the Container Storage Interface (CSI).

- **ContainerOperator**

Manages the container operations.

After installing the system and GUI package, you need to create the first GUI user to log in to the GUI. This user can create other GUI administrative users to perform system management and monitoring tasks. When you launch the GUI for the first time after the installation, the GUI welcome page provides options to create the first GUI user from the command line prompt by using the `/usr/lpp/mmfs/gui/cli/mkuser <user_name> -g SecurityAdmin` command.

User groups

Users who are part of *Security Administrator* and *User Administrator* user groups can create role-based user groups where any users that are added to the group adopt the role that is assigned to that group.

Roles apply to users on the system and are based on the user group to which the user belongs. A user can be part of multiple user groups so that a single user can play multiple roles in the system. You can assign the following roles to your user groups:

- **Administrator**

Users can access all functions on the GUI except those deals with managing users and user groups.

- **Security Administrator**

Users can access all functions on the GUI, including managing users and user groups.

- **System Administrator**

Users manage clusters, nodes, and alert logs.

- **Storage Administrator**

Users manage disks, file systems, pools, and filesets.

- **Snapshot Administrator**

Users manage snapshots for file systems, filesets.

- **Monitor**

Users can view objects and system configuration but cannot configure, modify, or manage the system or its resources.

- **Data Access**

Users can perform the following tasks:

- Edit owner, group, and ACL of any file or path through the **Files > File System ACL > Files and Directories** page.
- Edit owner, group, and ACL for a non-empty directory of a file system, fileset, NFS export, or SMB share.
- Create or delete object containers through the **Object > Accounts** page.

- **Protocol Administrator**

Users manage Object Storage and data export definitions of SMB and NFS protocols.

- **User Administrator**

Users manage GUI users and user groups.

- **CSI Administrator**

Users manage Container Storage Interface (CSI).

- **Container Operator**

Manages the container operations.

Note: Default groups are not created for the user role *User Administrator* in case the user is upgrading the IBM Storage Scale cluster from 4.2.0.x to a later release.

A default group is not created for the user role CSI Administrator in case the user is upgrading the IBM Storage Scale cluster from 5.0.3 or earlier.

For more information about how to create a GUI user and assign user roles, see “[Create GUI users and assign user permissions](#)” on page 453.

User repository

You can manage GUI users locally within the system and in an external authentication server such as Microsoft Active Directory (AD) or Lightweight Directory Access Protocol Server (LDAP).

Managing users locally in the IBM Storage Scale system

By default, the IBM Storage Scale system uses an internal authentication repository for GUI users. That is, the users who are created using the **Services > GUI** page are stored in the internal repository.

Managing GUI users in an external AD or LDAP server

By default, the IBM Storage Scale system uses an internal authentication repository for GUI users. You can configure an external authentication server either through GUI or CLI.

Note: You can configure external authentication only for GUI users who monitor and manage the cluster. The authentication method used for NFS and SMB users is different.

You can use the **Configure External Authentication** option that is available under the **External Authentication** tab to configure an external LDAP-based authentication method for authenticating the GUI users.

Use the **Test Connection** option that is available under the **External Authentication** tab to find out whether a user credential is available in the internal or external repository.

For more information about how to use the GUI to configure an external authentication method for the GUI users, see “[Configuring external authentication for GUI users](#)” on page 455.

Managing GUI user passwords

Use the various controls that are available under the **Password Policy** tab to enforce strong passwords for the GUI users whose credentials are stored in the internal repository.

Note: Only users with *UserAdmin* or *SecurityAdmin* role can modify the password policy of a user. If the password is expired for a user, the GUI logs off that user due to security reasons.

For more information about how to create password policy and to modify password, see the following topics:

- “[Defining a password policy for GUI users](#)” on page 454
- “[Changing or expiring password of GUI user](#)” on page 455

Create GUI users and assign user permissions

You can create users who can perform different administrative tasks on the system. Each user must be part of a user group or multiple groups that are defined on the system. When you create a new user, you assign the user to one of the default user groups or to a custom user group. User groups are assigned with predefined roles that authorize the users within that group to a specific set of operations on the GUI.

Note: Only users with *SecurityAdmin* or *UserAdmin* role can create a GUI user.

Perform the following steps to create a GUI user and assign user role:

1. Go to **Services > GUI** page in the IBM Storage Scale GUI.
2. Click **Groups**. The **Groups** tab lists all the user groups that are already available. You can either use the existing groups or create a group.
3. Click **Create Group** if you want to create a new group. The Create User Group window appears.
4. Type the new user group name in the **User group name** field.
5. Select a role for the new group from the list of user group roles.

For more information about the user groups and the user group roles, see [Chapter 35, “Managing GUI users,” on page 451](#).

6. Click **Create**. The new user group with the specified role is created.
7. Click **User**. The User section of the GUI page lists all the existing users and the associated user groups.
8. Click **Create User**. The Create User window appears.
9. Type the name of the new user in the **Name** field.
10. Select the user groups in which the user needs to be added. When you select the user group for a user, the permissions or roles that are defined for a user group becomes applicable to the user.
11. Type the password in the **Temporary password** and **Confirm password** fields.
12. Slide the button to enable the **Disable automatic password expiry** option. The user password will no longer expire and the user does not need to create a new password for every login session.
13. Click **Create**. A GUI user is created with certain user roles and permissions and added to the selected groups.

Defining a password policy for GUI users

You can enforce strong passwords for the GUI users whose credentials are stored in the internal repository by defining a password policy.

Note: Only users with *UserAdmin* or *SecurityAdmin* role can define password policy of a user.

Perform the following steps to a password policy for GUI users:

1. Go to **Services > GUI > Password Policy** page in the IBM Storage Scale GUI.
2. Select the password history size in the **Password history size** field. The password history size defines the number of unique new passwords associated with a user before an old password can be reused. To disable this feature, select 0.
3. Select the minimum number of alphanumeric characters required, in the **Minimum password length** field.
4. Select the minimum age of the password in the **Minimum password age** field. This age defines the minimum period for which the password can be used before it can be changed.
5. Select the maximum age of the password in the **Maximum password age** field. This age defines the maximum period for which the password can be used after which it must be changed.
6. Click **View advanced policies** and define the following details:
 - Minimum number of characters in the password
 - Number of uppercase alphabetic characters
 - Number of lowercase alphabetic characters
 - Number of numerical characters
 - Number of special characters
 - Number of consecutive repeating characters
 - Number of characters that must be changed in the new password
7. Select the **Disallow username in the password** checkbox if you do not want the user name to be used as the password.
8. Click **Save** to apply the password policy. A password policy is created for the GUI users.

Changing or expiring password of GUI user

Use the various controls that are available under the **Password Policy** tab of the **Services > GUI** page to enforce strong passwords for the GUI users. The GUI users credentials are stored in the internal repository.

Note: Only users with *UserAdmin* or *SecurityAdmin* role can modify the password policy of a user. If the user role or password is expired for a user, the GUI logs off that user due to security reasons.

Perform the following steps to change or expire password of a GUI user:

1. Go to **Services > GUI > Users** page in the IBM Storage Scale GUI.
2. Select the user for which you need to change the password.
3. Select **Edit Password** option that is available in the **Actions** menu. The **Modify Password** window appears.
4. Enter the new password in the **Modify Password** window.

Note: If you are using the internal authentication method for the GUI users, you can also use the **Edit Password** option that is available in the user menu, which is placed on the upper-right corner of the GUI. If you are using an external authentication method for the GUI users, the GUI does not display this option in the user menu.

5. To expire the password of a single user or all users, select **Expire Password** or **Expire Password of All Users** options from the **Actions** menu that is available under the **Users** tab. If the password is set as expired for a user, the user will be prompted to change the password in the next login.

Note: To ensure that the user password never expires, you can enable the **Disable automatic password expiry** option while creating the user. For more information, see “[Create GUI users and assign user permissions](#)” on page 453.

Configuring external authentication for GUI users

You can manage administrative users either locally within the system or in an external authentication server such as Microsoft Active Directory (AD) or Lightweight Directory Access Protocol Server (LDAP). By default, the IBM Storage Scale uses an internal authentication repository for administrative GUI users.

Perform the following steps to configure an external LDAP-based authentication method for authenticating the GUI users:

1. Go to **Services > GUI** page in the IBM Storage Scale GUI.
2. Click **External Authentication**.
3. Click **Configure External Authentication**. The **Configure LDAP-Based External Authentication** wizard appears.
4. Select an external authentication repository from the **Repository type** field.

You can store the user credentials in the following repository types:

- Microsoft Active Directory
- IBM Lotus® Domino®
- IBM SecureWay Directory Server
- IBM Tivoli Directory Server
- Netscape Directory Server
- Novell eDirectory
- Sun Java™ System Directory Server
- Custom

Note: This procedure explains how to configure the authentication method by using Microsoft Active Directory as the repository type.

5. Click **Next**.

6. Type the LDAP server IP address or host name in the **Server** field.
 7. Type the port number in the **Port** field.
 8. Specify the BaseDN string for the repository in the **Base DN** field.
 9. Specify the BindDN string that is used for searching the authentication user, in the **Bind DN** field.
 10. Type the password of the authentication user in the **Bind DN** password field.
 11. Select the **Use SSL certificate** checkbox if you want to use an SSL certificate to secure the connection between the GUI server and the external authentication server. If you select this option, upload the keystore file and type the keystore password in the **Keystore password** field.
 12. Select the **Use truststore** checkbox. If you select this option, upload the truststore file and type the truststore password in the **Truststore password** field.
- Note:** The truststore file is located at /opt/IBM/wlp/usr/servers/gpfsgui/resources/security.
13. Click **Next**. The **Search Settings** page of the Configure LDAP-Based External Authentication wizard appears. The fields are already populated with default values from the Account Name template. If necessary, you can modify the values.
 14. Type the filter clause for searching the user registry for users, in the **User filter** field.
 15. Type the filter that maps the name of a user to an LDAP entry, in the **User ID** map field.
 16. Type the filter clause for searching the user registry for groups, in the **Group filter** field.
 17. Type the filter that maps the name of a group to an LDAP entry, in the **Group ID** map field.
 18. Specify the filter that identifies user to group memberships, in the **Group member ID map** field.
 19. Select the **Use recursive search** checkbox to enable the nested group search.

- Note:** A recursive search identifies all the nested groups that are mapped to the GUI group to which an external LDAP user belongs. GUI authentication works for all users who are assigned to either the sub-groups or the main group of the nested group structure.
20. Apply the **User Principal Name** template if required. Accordingly, the values that you entered in the various fields of the **Search Settings** page of the wizard change.
 21. Click **Next** after making changes. The **Summary** page of the wizard appears.
 22. Review the configuration details and click **Finish**.

An LDAP-based external authentication method is configured for the GUI users. Configuring an external authentication method for the GUI users prompts the system to log out the already logged-in GUI users. You need to log in to the system again.

You can log in to the IBM Storage Scale GUI and create group mappings through the GUI on the **Services > GUI > Users** page by using the **Create Group Mapping** option.

You can edit or delete the external authentication by using the **Edit** and **Delete** options that are available in the **Services > GUI > External Authentication** page of the GUI.

Configuring external authentication by using CLI

Perform the following steps to configure external authentication by using CLI:

1. Create your AD or LDAP configuration by issuing the **mkldap** command at the following location: /usr/lpp/mmfs/gui/cli/mkldap.

This command writes the configuration automatically to /opt/ibm/wlp/usr/servers/gpfsgui/ldap.xml, which is then distributed across all GUI nodes. For secure AD or LDAP connection, make sure that the keystores are present on the respective GUI nodes.

The **mkldap** command accepts the following parameters:

Table 33. ***mkldap*** command parameters

Parameters	Description
<code>id</code>	Unique ID of the LDAP configuration.
<code>--host</code>	The IP address or host name of the LDAP server.
<code>--baseDn</code>	BaseDn string for the repository.
<code>--bindDn</code>	BindDn string for the authentication user.
<code>--bindPassword</code>	Password of the authentication user.
<code>--port</code>	Port number of the LDAP. Default is 389 or 636 over SSL.
<code>--type</code>	Repository type such as " <i>Microsoft Active Directory, ids, domino, secureway, iplanet, netscape, edirectory</i> " or " <i>custom</i> ". Default value is " <i>Microsoft Active Directory</i> ".
<code>--connectTimeout</code>	Maximum time for establishing a connection with the LDAP server. Default value is 1 m.
<code>--searchTimeout</code>	Maximum time for an LDAP server to respond before a request is canceled. Default value is 1 m.
<code>--keystore</code>	Location with file name of the keystore file (.jks, .p12 or .pfx).
<code>--keystorePassword</code>	Password of the keystore.
<code>--truststore</code>	Location with file name of the truststore file (.jks, .p12 or .pfx).
<code>--truststorePassword</code>	Password of the truststore.
<code>--userFilter</code>	User filter for the LDAP repository.
<code>--userIdMap</code>	User ID map for the LDAP repository.
<code>--groupFilter</code>	Group filter for the LDAP repository.
<code>--groupIdMap</code>	Group ID map for the LDAP repository.
<code>--groupMemberIdMap</code>	Group member ID map for the LDAP repository.

Example for standard AD

```
mkldap myad --host 9.155.106.19 --bindDn CN=Administrator,CN=Users,DC=mydomain,DC=local  
--baseDn CN=Users,DC=mydomain,DC=local
```

Example for secure AD

```
mkldap mysecuread --host 9.155.106.19 --bindDn  
CN=Administrator,CN=Users,DC=mydomain,DC=local  
--baseDn CN=Users,DC=mydomain,DC=local --keystore /tmp/ad.jks
```

If you specify multiple AD or LDAP servers, you might encounter a problem that a user with the same user name exists in multiple user repositories. This user cannot be able to log in. To prevent this situation, you can specify LDAP filters for User Principal Names (UPN) for a selected server configuration.

Example for a scenario where UPN filters are enabled

```
mkldap myfilteredad --host 9.155.106.19 --bindDn  
CN=Administrator,CN=Users,DC=mydomain,DC=local  
--baseDn CN=Users,DC=mydomain,DC=local --userFilter "(&(userPrincipalName=%v)  
(objectcategory=person))"  
--groupFilter "(&(cn=%v)(objectcategory=group))" --userIdMap "*:userPrincipalName"  
--groupIdMap "*:cn" --groupMemberIdMap "memberOf:member"
```

2. Map an existing AD or LDAP group to the *SecurityAdmin* GUI role as shown in the following example:

```
/usr/lpp/mmfs/gui/cli/mkusergrp LDAPGroup --role securityadmin
```

Now you can log in with your AD or LDAP user and create more group mappings through the GUI on the **Services > GUI > Users** page by using the **Create Group Mapping** option.

If you want to remove the existing configurations, use the **rmldap** command. To see all specified LDAP configurations, issue the **lsldap** command.

Note: Configurations that are managed by **mkldap** and **rmldap** commands are not overwritten during the upgrade. That is you do not need to back up the configuration data.

Configuring multi-factor authentication for GUI users

You can configure multi-factor authentication for GUI users who are registered in IBM Security Verify repository. All users eligible for this feature are able to provide the additional authentication credentials that they receive on their registered email address or phone number and log in to the GUI.

The following steps help you configure multi-factor authentication method for authenticating the GUI users:

Note: If you are unable to view this feature, it indicates that Multi-Factor Authentication is not configured with your account. Contact your System Administrator for more details.

1. Go to **Services > GUI** page in the IBM Storage Scale GUI.
2. Click **Additional Authentication**.
3. Click **Configure Multi-Factor Authentication**. The **Configure Multi-Factor Authentication** window appears.
4. To enable the GUI for multi-factor authentication, use the **GUI ENABLED** slider.
5. Type the proxy server name and the port number it uses to connect to the GUI in **Proxy server**.
9.155.107.11:8080
- Note:** It is not mandatory to provide the proxy server name and port number.
6. Type your ID that you have used to configure the multi-factor authentication, in the **Client ID** field.
7. Type the hostname that is added in the IBM Security Verify server, in the **Tenant name** field. For more information, see [“Configuring GUI details in IBM Security Verify for multi-factor authentication” on page 459](#).

8. Type the secret information that you have added in the IBM Security Verify server, in the **Client secret** field.
9. Click **Test Connection** to check whether your selected user is enabled for multi-factor authentication.
10. Click **Configure** to complete the configuration of the multi-factor authentication.

On completing the configuration, you are prompted to log out of the system.

You can log in to the IBM Storage Scale GUI again and create groups by using the **Services > GUI > Groups** option and assign users to those groups by using the **Services > GUI > Users** option. You can enable multi-factor authentication for all new users.

Note: You cannot enable multi-factor authentication for User Groups that use inter-service REST API based communication.

Configuring GUI details in IBM Security Verify for multi-factor authentication

You can configure the GUI details including the tenant name, Client ID, secret, and user roles in IBM Security Verify (ISV) to enable the multi-factor authentication feature through ISV. As a first step, you need to create an account in ISV that authenticates your credentials and verifies your email before providing you with access to configure the tenant and client details.

Follow the procedure to configure the GUI client for multi-factor authentication with IBM Security Verify:

1. Create an account at the following URL:

```
https://www.ibm.com/in-en/products/verify-for-workforce-iam
```

2. Click **Try free edition**.
3. On the **Set up your tenant** page, type the tenant name.

Note: Copy the tenant name. You need it for configuring multi-factor authentication in IBM Storage Scale GUI.

4. Click **Create tenant**.
An email notification from the ISV team confirms that your account is successfully created.
5. Click **Go To IBM Security Verify** in the email that you received. The IBM Security Verify GUI is displayed.
6. On the **Welcome** page, agree to the terms and conditions.
7. From the navigation menu, click **Security > API access**.
8. Click **Add API client**.
9. On the **Create API client** page, choose the relevant entitlements for the client you are configuring.

Note: You can use the **Select all** checkbox to select all the listed entitlements.

10. Click **Next**.
11. On the **Custom scope** page, select **Allow configured scopes only** to define scopes to limit access to the access tokens.
12. Click **Next**.
13. On the **IP filter** page, select **Enable IP filtering** to limit token creation requests to a specific range of IP addresses.
14. Click **Next**.
15. On the **Additional properties** page, provide any additional attributes that you need to define for the client.

Note: Steps 11 - 13 are optional steps.

16. Click **Next**.
17. On the **Confirm configuration** page, type the client name and provide a description, if necessary. For example, scale-gui.
18. Click **Create API client**.

19. From the navigation menu, click **Security > API access** and select the API client that you have added in step 17.

20. Click  to view the configuration details.

21. From the **Configuration** list copy the **Client ID** and **Client secret** that are automatically generated when the client is created and are available under the **API credentials** section.

22. From the navigation menu, select **Directory > Users & groups** and then click **Add user**.

23. On the **Add user** page, create a user and their related information that includes their mobile number and email address.

Note:

- Mobile number is necessary only if you want to enable mobile OTP as an authentication option for the user. If required, the user can change this number later. It is important to provide a number with a valid country code.
- The username that you add here must be the same as the one configured in IBM Storage Scale GUI.

24. Click **Save**.

Your GUI client is now configured and GUI user is successfully added in the ISV repository.

Chapter 36. Managing GPFS access control lists

Access control protects directories and files by providing a means of specifying who is granted access. GPFS access control lists (ACL) are either traditional ACLs based on the POSIX model, or NFS V4 ACLs. NFS V4 ACLs are different from traditional ACLs, and provide improved control of file and directory access. A GPFS file system can also be exported by using NFS.

Management of GPFS access control lists (ACLs) and NFS export includes the following topics:

- [“Traditional GPFS ACL administration” on page 461](#)
- [“NFS V4 ACL administration” on page 465](#)
- [Chapter 37, “Native NFS and GPFS,” on page 499](#)

Note: The **cp --preserve=xattr** Linux command copies either the POSIX or the NFSv4 ACL extended attributes when an IBM Storage Scale file is copied. Also, the following system calls are extended when they are applied to files in IBM Storage Scale file systems:

- The **listxattr()** system call, lists the attributes that represent the POSIX or NFSv4 ACL.
- The **getxattr()** system call, retrieves the specified POSIX or NFSv4 ACL attribute. The content of the ACL is retrieved in the **system posix_acl_access** attribute or the **system gpfs_nfs4_acl** attribute.
- The **setxattr()** system call, writes the content of the specified POSIX or NFSv4 ACL attribute to the corresponding ACL.

In versions of IBM Storage Scale earlier than 5.0.5, neither POSIX nor NFSv4 ACLs are supported in this way. However, it is possible to copy the POSIX ACL by issuing the **cp --preserve=mode** command.

Traditional GPFS ACL administration

Support for NFS V4 access control lists (ACLs) is added to traditional ACL support. NFS V4 ACLs are different than the traditional ones.

If you are using NFS V4 ACLs, see [“NFS V4 ACL administration” on page 465](#). Both ACL types can coexist in a single GPFS file system.

Traditional GPFS ACLs are based on the POSIX model. Traditional GPFS access control lists (ACLs) extend the base permissions, or standard file access modes of read (r), write (w), and execute (x). The permissions are extended beyond the three categories of file owner, file group, and other users. This extension of permissions allows the definition of more users and user groups. In addition, GPFS introduces a fourth access mode, control (c), which can be used to govern who can manage the ACL itself.

In this way, a traditional ACL can be created that looks like this:

```
#owner:jesmith
#group:team_A
user::rwx
group::rwx-
other::--x-
mask::rwx
user:alpha:r-xc
group:audit:r-x-
group:system:rwx-
```

In this ACL:

- The first two lines are comments that show the file's owner, jesmith, and group name, team_A.
- The next three lines contain the base permissions for the file. These three entries are the minimum necessary for a GPFS ACL:
 1. The permissions set for the file owner (user), jesmith
 2. The permissions set for the owner's group, team_A

- 3. The permissions set for other groups or users outside the owner's group and not belonging to any named entry
- The next line, with an entry type of mask, contains the maximum permissions that are allowed for any entries other than the owner (the user entry) and those that are covered by other in the ACL.
- The last three lines contain additional entries for specific users and groups. These permissions are limited by those specified in the mask entry, but you can specify any number of additional entries up to a memory page (approximately 4 K) in size.

Traditional GPFS ACLs are fully compatible with the base operating system permission set. Any change to the base permissions by using the chmod command, for example, modifies the corresponding GPFS ACL as well. Similarly, any change to the GPFS ACL is reflected in the output of commands such as ls -l. The control (c) permission is GPFS-specific. There is no comparable support in the base operating system commands. As a result, the (c) permission is visible only with the GPFS ACL commands.

Each GPFS file or directory has an *access ACL* that determines its access privileges. These ACLs control who is allowed to read or write at the file or directory level. The ACLs also control who can change the ACL itself.

In addition to an *access ACL*, a directory might also have a *default ACL*. If present, the default ACL is used as a base for the access ACL of every object that is created in that directory. This allows a user to protect all files in a directory without explicitly setting an ACL for each one.

When a new object is created, and the parent directory has a default ACL, the entries of the default ACL are copied to the new object's access ACL. Then, the base permissions for user, mask (or group if mask is not defined), and other, are changed to their intersection. This change takes place with the corresponding permissions from the mode parameter in the function that creates the object.

If the new object is a directory, its default ACL is set to the default ACL of the parent directory. If the parent directory does not have a default ACL, the initial access ACL of newly created objects consists only of the three required entries (user, group, other). The values of these entries are based on the mode parameter in the function that creates the object and the umask currently in effect for the process.

Administrative tasks that are associated with traditional GPFS ACLs are:

1. [“Setting traditional GPFS access control lists” on page 462](#)
2. [“Displaying traditional GPFS access control lists” on page 463](#)
3. [“Changing traditional GPFS access control lists” on page 465](#)
4. [“Deleting traditional GPFS access control lists” on page 465](#)

Related concepts

NFS V4 ACL administration

AIX does not allow a file system to be NFS V4 exported unless it supports NFS V4 ACLs. By contrast, Linux does not allow a file system to be NFS V4 exported unless it supports POSIX ACLs.

Authorizing protocol users

Authorization grants or denies access to resources such as directories, files, commands, and functions. Authorization is applicable to an already authenticated identity, such as an IBM Storage Scale data user, an administrative user, or an IBM service representative. Access to the files and directories of the IBM Storage Scale system is managed through Access Control Lists (ACLs). It ensures that only authorized users get access to exports, directories, and files. An Access Control Entry (ACE) is an individual entry in an access control list, and describes the permissions for an individual user or group of users. An ACL can have zero or more ACEs.

Setting traditional GPFS access control lists

Use the following information to set GPFS access control lists (ACLs).

GUI navigation

To work with this function in the GUI, log on to the IBM Storage Scale GUI and select **Access > File System ACL**.

Use the `mmputacl` command to set the access ACL of a file or subdirectory, or the default ACL of a directory. For example, to set the ACL for a file named `project2.history`, you can create a file that is named `project2.acl` that contains:

```
user::rwxc
group::rwx-
other::--x-
mask::rwxc
user:alpha:r-xc
group:audit:rw--
group:system:rwx-
```

In this example,

- The first three lines are the required ACL entries that set the permissions for the file's owner, the owner's group, and for processes that are not covered by any other ACL entry.
- The last three lines contain named entries for specific users and groups.
- Because the ACL contains named entries for specific users and groups, the fourth line contains the required mask entry, which is applied to all named entries (entries other than the `user` and `other`).

After you are satisfied that the correct permissions are set in the ACL file, you can apply them to the target file with the `mmputacl` command. For example, to set permissions contained in the file `project2.acl` for the file `project2.history`, enter:

```
mmputacl -i project2.acl project2.history
```

To confirm the changes, enter:

```
mmgetacl project2.history
```

The information sent to standard output is similar to:

```
#owner:guest
#group:usr
user::rwxc
group::rwx- #effective:rw--
other::--x-
mask::rw-c
user:alpha:rwxc #effective:rw-c
group:audit:rwx- #effective:rw--
group:system:-w--
```

You can issue the `mmputacl` command without using the `-i` option to specify an ACL input file, and make ACL entries through standard input. However, the `-i` option is more useful for avoiding errors when you are creating a new ACL.

For complete usage information, see the *mmputacl command* and the *mmgetacl command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related tasks

[Displaying traditional GPFS access control lists](#)

[Applying an existing traditional GPFS access control list](#)

[Changing traditional GPFS access control lists](#)

[Deleting traditional GPFS access control lists](#)

Displaying traditional GPFS access control lists

Use the `mmgetacl` command to display the access ACL of a file or subdirectory, or the default ACL of a directory. For example, to display the ACL for the file `project2.history`, enter:

```
mmgetacl project2.history
```

The information sent to standard output is similar to:

```
#owner:guest
#group:usr
user::rwx
group::rwx- #effective:rw-
other::--x-
mask::rw-
user:alpha:rwx #effective:rw-c
group:audit:rwx- #effective:rw-
group:system:-w-
```

The first two lines are comments that are displayed by the `mmgetacl` command, showing the owner and owning group. All entries containing permissions that are not allowed (because they are not set in the mask entry) display a comment that shows their effective permissions.

For complete usage information, see the *mmgetacl command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related tasks

[Setting traditional GPFS access control lists](#)

Use the following information to set GPFS access control lists (ACLs).

[Applying an existing traditional GPFS access control list](#)

[Changing traditional GPFS access control lists](#)

[Deleting traditional GPFS access control lists](#)

Applying an existing traditional GPFS access control list

To apply the same traditional ACLs from one file or directory to another:

1. Issue the `mmgetacl` command with the `-o` option to place the information in an output file.
2. Apply the ACLs to the new file or directory by issuing the `mmpputacl` command with the `-i` option.

For example, use the `-o` option to specify a file to which the ACL is written:

```
mmgetacl -o old.acl project2.history
```

Then, to assign the same permissions to another file, `project.notes`, enter:

```
mmpputacl -i old.acl project.notes
```

To confirm the changes, enter:

```
mmgetacl project.notes
```

The information sent to standard output is similar to:

```
#owner:guest
#group:usr
user::rwx
group::rwx- #effective:rw-
other::--x-
mask::rw-
user:alpha:rwx #effective:rw-c
group:audit:rwx- #effective:rw-
group:system:-w-
```

For complete usage information, see the *mmgetacl command* and the *mmpputacl command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related tasks

[Setting traditional GPFS access control lists](#)

Use the following information to set GPFS access control lists (ACLs).

[Displaying traditional GPFS access control lists](#)

[Changing traditional GPFS access control lists](#)

[Deleting traditional GPFS access control lists](#)

Changing traditional GPFS access control lists

Use the `mmeditacl` command to change or create the traditional ACL of a file or directory, or the default ACL of a directory. For example, to interactively edit the ACL for the file `project2.history`, enter:

```
mmeditacl project2.history
```

The current ACL entries are displayed by using the default editor, if the `EDITOR` environment variable specifies a complete path name. When the file is saved, the system displays information similar to:

```
mmeditacl: 6027-967 Should the modified ACL be applied? (yes) or (no)
```

After you respond yes, the ACLs are applied.

For complete usage information, see the *mmeditacl command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related tasks

[Setting traditional GPFS access control lists](#)

Use the following information to set GPFS access control lists (ACLs).

[Displaying traditional GPFS access control lists](#)

[Applying an existing traditional GPFS access control list](#)

[Deleting traditional GPFS access control lists](#)

Deleting traditional GPFS access control lists

Use the `mmdelacl` command to delete the extended entries in a traditional ACL of a file or directory, or the default ACL of a directory. For example, to delete the ACL for the directory `project2`, enter:

```
mmdelacl project2
```

To confirm the deletion, enter:

```
mmgetacl project2
```

The system displays information similar to:

```
#owner:uno
#group:system
user::rwx
group::r-x-
other::---x-
```

You cannot delete the base permissions, which remain in effect after this command is executed.

For complete usage information, see the *mmdelacl command* and the *mmgetacl command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related tasks

[Setting traditional GPFS access control lists](#)

Use the following information to set GPFS access control lists (ACLs).

[Displaying traditional GPFS access control lists](#)

[Applying an existing traditional GPFS access control list](#)

[Changing traditional GPFS access control lists](#)

NFS V4 ACL administration

AIX does not allow a file system to be NFS V4 exported unless it supports NFS V4 ACLs. By contrast, Linux does not allow a file system to be NFS V4 exported unless it supports POSIX ACLs.

This is because NFS V4 Linux servers handle NFS V4 ACLs by translating them into POSIX ACLs. For more information, see “[Linux ACLs and extended attributes](#)” on page 474.

Note:

This topic applies only to kernel NFS and does not refer to the NFS Server function included with CES. For more information, see “[Authorizing protocol users](#)” on page 474.

With AIX, the file system must be configured to support NFS V4 ACLs (with the `-k all` or `-k nfs4` option of the `mmcrfs` or `mmchfs` command). The default for the `mmcrfs` command is `-k all`.

With Linux, the file system must be configured to support POSIX ACLs (with the **-k all** or **-k posix** option of the `mmcrfs` or `mmchfs` command).

Depending on the value (`posix | nfs4 | all`) of the `-k` parameter, one or both ACL types can be allowed for a given file system. Since ACLs are assigned on a per-file basis, this means that within the same file system one file can have an NFS V4 ACL, while another has a POSIX ACL. The type of ACL can be changed by using the `mmpuac1` or `mmeditac1` command to assign a new ACL. You can also change the type of ACL by the `mmdelac1` command (causing the permissions to revert to the mode, which is in effect a POSIX ACL). At any point in time, only a single ACL can be associated with a file. Access evaluation is done as required by the ACL type that is associated with the file.

NFS V4 ACLs are represented in a different format than traditional ACLs. For detailed information on NFS V4 and its ACLs, refer to *NFS Version 4 Protocol* and other information that is found in the [Network File System Version 4 \(nfsv4\)](#) section of the IETF Datatracker website (datatracker.ietf.org/wg/nfsv4/documents).

The concept of a default ACL does not exist for NFS V4 ACLs. Instead, there is a single ACL and the individual ACL entries can be flagged as being *inherited* (either by files, directories, both, or neither). Therefore, specifying the `-d` flag on the `mmpuac1` command for an NFS V4 ACL is an error.

Related concepts

[Traditional GPFS ACL administration](#)

Support for NFS V4 access control lists (ACLs) is added to traditional ACL support. NFS V4 ACLs are different than the traditional ones.

[Authorizing protocol users](#)

Authorization grants or denies access to resources such as directories, files, commands, and functions. Authorization is applicable to an already authenticated identity, such as an IBM Storage Scale data user, an administrative user, or an IBM service representative. Access to the files and directories of the IBM Storage Scale system is managed through Access Control Lists (ACLs). It ensures that only authorized users get access to exports, directories, and files. An Access Control Entry (ACE) is an individual entry in an access control list, and describes the permissions for an individual user or group of users. An ACL can have zero or more ACEs.

NFS V4 ACL Syntax

An NFS V4 ACL consists of a list of ACL entries. Where traditional ACLs can display one entry per line, the GPFS representation of NFS V4 ACL entries is of three lines each, due to the increased number of available permissions beyond the traditional `rwx`.

The first line separates the multiple parts by colons (':').

- The first part identifies the user or group.
- The second part displays a `rwx` translation of the permissions that appear on the subsequent two lines.
- The third part is the ACL type. NFS V4 provides both an *allow* and *deny* type.

allow

Means to allow (or permit) those permissions that are selected with an 'X'.

deny

Means to not allow (or deny) those permissions that are selected with an 'X'.

- The fourth and final part is a list of flags that indicate *inheritance*.

Valid flag values are:

DirInherit

Indicates that the ACL entry must be included in the initial ACL for subdirectories that are created in this directory (and the current directory).

FileInherit

Indicates that the ACL entry must be included in the initial ACL for files that are created in this directory.

Inherited

Indicates that the current ACL entry was derived from inherit entries in an NFS v4 ACL of the parent directory.

InheritOnly

Indicates that the current ACL entry must *not* apply to the directory, but *must* be included in the initial ACL for objects that are created in this directory.

NoPropagateInherit

Indicates that the ACL entry must be included in the initial ACL for subdirectories that are created in this directory but not further propagated to subdirectories created below *that* level.

As in traditional ACLs, users and groups are identified by specifying the type and name. For example, `group:staff` or `user:bin`. NFS V4 provides for a set of special names that are not associated with a specific local UID or GID. These special names are identified with the keyword `special` followed by the NFS V4 name. These names are recognized by the fact that they end with the character '@'. For example, `special:owner@` refers to the owner of the file, `special:group@` the owning group, and `special:everyone@` applies to all users.

The next two lines provide a list of the available access permissions that can be *allowed* or *denied*, based on the ACL type specified on the first line. A permission is selected by using an 'X'. Permissions that are not specified by the entry must be left marked with '-' (minus sign).

Starting from IBM Storage Scale 5.1.7, IBM Storage Scale supports setting the extended `system.nfs4_acl` attribute as another method for manipulating NFSv4 ACLs. This enhancement is added to support the Linux NFSv4 ACL command-line tools. You can employ the syntax of these tools to manage NFSv4 ACLs in IBM Storage Scale. For requirements and limitations, see [Q.2.41 in IBM Storage Scale FAQ](#).

These are examples of NFS V4 ACLs.

1. An ACL entry that explicitly allows READ, EXECUTE and READ_ATTR to the staff group on a file is similar to this:

```
group:staff:r-x::allow
  (X)READ/LIST (-)WRITE/CREATE (-)APPEND/MKDIR (-)SYNCHRONIZE (-)READ_ACL  (X)READ_ATTR
  (-)READ_NAMED
  (-)DELETE   (-)DELETE_CHILD (-)CHOWN (X)EXEC/SEARCH (-)WRITE_ACL (-)WRITE_ATTR
  (-)WRITE_NAMED
```

2. A Directory ACL is similar to this. It can include *inherit* ACL entries that do not apply to the directory itself, but instead become the initial ACL for any objects that are created within the directory.

```
special:group@:----:deny:DirInherit:InheritOnly
  (X)READ/LIST (-)WRITE/CREATE (-)APPEND/MKDIR (-)SYNCHRONIZE (-)READ_ACL  (X)READ_ATTR
  (-)READ_NAMED
  (-)DELETE   (-)DELETE_CHILD (-)CHOWN (X)EXEC/SEARCH (-)WRITE_ACL (-)WRITE_ATTR
  (-)WRITE_NAMED
```

3. A complete NFS V4 ACL is similar to this:

```
#NFSv4 ACL
#owner:smithj
#group:staff
special:owner@:rwx:allow:FileInherit
  (X)READ/LIST (X)WRITE/CREATE (X)APPEND/MKDIR (-)SYNCHRONIZE (X)READ_ACL  (X)READ_ATTR
  (-)READ_NAMED
  (-)DELETE   (X)DELETE_CHILD (X)CHOWN (X)EXEC/SEARCH (X)WRITE_ACL (X)WRITE_ATTR
  (-)WRITE_NAMED

special:owner@:rwx:allow:DirInherit:InheritOnly
```

```

(X)READ/LIST (X)WRITE/CREATE (X)APPEND/MKDIR (-)SYNCHRONIZE (X)READ_ACL (X)READ_ATTR
(-)READ_NAMED
(X)DELETE (X)DELETE_CHILD (X)CHOWN (X)EXEC/SEARCH (X)WRITE_ACL (-)WRITE_ATTR
(-)WRITE_NAMED

user:smithj:rwx:allow
(X)READ/LIST (X)WRITE/CREATE (X)APPEND/MKDIR (-)SYNCHRONIZE (X)READ_ACL (X)READ_ATTR
(-)READ_NAMED
(X)DELETE (X)DELETE_CHILD (X)CHOWN (X)EXEC/SEARCH (X)WRITE_ACL (-)WRITE_ATTR
(-)WRITE_NAMED

```

Note: In IBM Storage Scale 5.0.3, a difference in the handling of the NFSv4 ACL bit SYNCHRONIZE can cause access issues for Microsoft Windows clients. The change is that when ACL data is returned to the SMB client, the SYNCHRONIZE bit on ACL "allow" entries is passed unchanged. But Microsoft Windows clients require the SYNCHRONIZE bit to be set for renaming files or directories. Files that are written by Microsoft Windows clients usually have the SYNCHRONIZE bit set.

To restore the pre-5.0.3 behavior, issue the following command for each SMB share that is affected by the problem:

```
/usr/lpp/mmfs/bin/net conf setparm <SMBShareName> 'nfs4:set synchronize' yes
```

In the long term, it is a good idea to change the ACLs for all files and directories that are missing the SYNCHRONIZE bit instead of modifying the SMB configuration.

Related concepts

[NFS V4 ACL translation](#)

[Considerations when using GPFS with NFS V4 ACLs](#)

Related tasks

[Setting NFS V4 access control lists](#)

[Displaying NFS V4 access control lists](#)

[Applying an existing NFS V4 access control list](#)

[Changing NFS V4 access control lists](#)

[Deleting NFS V4 access control lists](#)

Related reference

[Exceptions and limitations to NFS V4 ACLs support](#)

Review the exceptions and limitations to NFS V4 ACLs in IBM Storage Scale.

ACL entries **DELETE** and **DELETE_CHILD**

The ACL entries DELETE and DELETE_CHILD require special considerations. The effect of various combinations of the DELETE attribute for a file, and the DELETE_CHILD attribute for its parent directory, is given in [Table 34 on page 469](#).

In this table, the columns refer to the ACL entry for a given file, and the rows refer to the ACL entry for its parent directory. The various combinations of these attributes produce one of these results:

Permit

Indicates that GPFS permits removal of a file with the combination of file and parent directory ACL entries specified. (Other permission checking can exist within the operating system as well.)

Deny

Indicates that GPFS denies (does not permit) removal of a file with the combination of file and parent directory ACL entries specified.

Removal of a file includes renaming the file, moving the file from one directory to another even if the file name remains the same, and deleting it.

Table 34. Removal of a file with ACL entries DELETE and DELETE_CHILD

	ACL Allows DELETE	ACL Denies DELETE	DELETE not specified	UNIX mode bits only
ACL Allows DELETE_CHILD	Permit	Permit	Permit	Permit
ACL Denies DELETE_CHILD	Permit	Deny	Deny	Deny
DELETE_CHILD not specified	Permit	Deny	Deny	Deny
UNIX mode bits only - wx permissions allowed	Permit	Permit	Permit	Permit
UNIX mode bits only - no w or no x permissions allowed	Permit	Deny	Deny	Deny

The UNIX mode bits are used in cases where the ACL is not an NFS V4 ACL.

NFS V4 ACL translation

NFS V4 access requires that an NFS V4 ACL is returned to clients whenever the ACL is read. This means that if a traditional GPFS ACL is associated with the file, a translation to NFS V4 ACL format must be performed when the ACL is read by an NFS V4 client. Since this translation must be done, an option (-k nfs4) is provided on the mmgetacl and mmeditacl commands so that this translation can be seen locally as well.

It can also be the case that NFS V4 ACLs are set for some file system objects (directories and individual files) before the administrator action to revert to a POSIX-only configuration. Since the NFS V4 access evaluation is no longer performed, it is desirable that the mmgetacl command returns an ACL representative of the evaluation that now occurs (translating NFS V4 ACLs into traditional POSIX style). The -k posix option returns the result of this translation.

Users can see ACLs in their true form. They can also see how they are translated for access evaluations. There are four cases:

1. By default, the mmgetacl command returns the ACL in a format consistent with the file system setting:
 - If posix only, it is shown as a traditional ACL.
 - If nfs4 only, it is shown as an NFS V4 ACL.
 - If all formats are supported, the ACL is returned in its true form.
2. The command mmgetacl -k nfs4 always produces an NFS V4 ACL.
3. The command mmgetacl -k posix always produces a traditional ACL.
4. The command mmgetacl -k native always shows the ACL in its true form, regardless of the file system setting.

In general, users must continue to use the mmgetacl and mmeditacl commands without the -k flag, allowing the ACL to be presented in a form appropriate for the file system setting. The NFS V4 ACLs are more complicated and hence harder to construct initially. Therefore, users who want to assign an NFS V4 ACL must use the command mmeditacl -k nfs4 to start with a translation of the current ACL. They can then modify the NFS V4 ACL that is returned.

Starting from IBM Storage Scale 5.1.7, IBM Storage Scale supports setting the extended **system.nfs4_acl** attribute as another method for manipulating NFSv4 ACLs. This enhancement is added to support the Linux NFSv4 ACL command-line tools. The **nfs4_getfac1** and **nfs4_setfac1** commands can be used directly in IBM Storage Scale to get and set NFSv4 ACLs. For requirements and limitations, see [Q.2.41 in IBM Storage Scale FAQ](#).

Related concepts

[NFS V4 ACL Syntax](#)

[Considerations when using GPFS with NFS V4 ACLs](#)

Related tasks

[Setting NFS V4 access control lists](#)

[Displaying NFS V4 access control lists](#)

[Applying an existing NFS V4 access control list](#)

[Changing NFS V4 access control lists](#)

[Deleting NFS V4 access control lists](#)

Related reference

[Exceptions and limitations to NFS V4 ACLs support](#)

Review the exceptions and limitations to NFS V4 ACLs in IBM Storage Scale.

Setting NFS V4 access control lists

There is no option on the `mmputacl` command to identify the type (traditional or NFS V4) of ACL that is to be assigned to a file. Instead, the ACL is assumed to be in the traditional format unless the first line of the ACL is:

```
#NFSv4 ACL
```

The lines that follow the first one are then processed according to the rules of the expected ACL type.

An NFS V4 ACL is similar to the sample shown:

```
#NFSv4 ACL
#owner:root
#group:system
special:owner@:rwx:allow
(X)READ/LIST (X)WRITE/CREATE (-)APPEND/MKDIR (X)SYNCHRONIZE (X)READ_ACL (-)READ_ATTR
(-)READ_NAMED
(X)DELETE (-)DELETE_CHILD (-)CHOWN (X)EXEC/SEARCH (X)WRITE_ACL (X)WRITE_ATTR (-)WRITE_NAMED

special:owner@:----:deny
(-)READ/LIST (-)WRITE/CREATE (-)APPEND/MKDIR (-)SYNCHRONIZE (-)READ_ACL (-)READ_ATTR
(X)READ_NAMED
(-)DELETE (X)DELETE_CHILD (X)CHOWN (-)EXEC/SEARCH (-)WRITE_ACL (-)WRITE_ATTR (X)WRITE_NAMED

user:guest:r-xc:allow
(X)READ/LIST (-)WRITE/CREATE (-)APPEND/MKDIR (X)SYNCHRONIZE (X)READ_ACL (-)READ_ATTR
(-)READ_NAMED
(X)DELETE (-)DELETE_CHILD (-)CHOWN (X)EXEC/SEARCH (X)WRITE_ACL (-)WRITE_ATTR (-)WRITE_NAMED

user:guest:----:deny
(-)READ/LIST (-)WRITE/CREATE (-)APPEND/MKDIR (-)SYNCHRONIZE (-)READ_ACL (-)READ_ATTR
(X)READ_NAMED
(-)DELETE (X)DELETE_CHILD (X)CHOWN (-)EXEC/SEARCH (-)WRITE_ACL (X)WRITE_ATTR (X)WRITE_NAMED
```

This ACL shows four ACL entries (an allow and deny entry for each of `owner@` and `guest`).

In general, constructing NFS V4 ACLs is more complicated than traditional ACLs. Users new to NFS V4 ACLs can find it useful to start with a traditional ACL. They can allow either `mmgetacl` or `mmeditacl` to provide the NFS V4 translation, by using the `-k nfs4` flag as a starting point when creating an ACL for a new file.

Starting from IBM Storage Scale 5.1.7, IBM Storage Scale supports setting the extended `system.nfs4_acl` attribute as another method for manipulating NFSv4 ACLs. This enhancement is added to support the Linux NFSv4 ACL command-line tools. The `nfs4_setfac1` command can be used directly in IBM Storage Scale to set NFSv4 ACLs. For requirements and limitations, see [Q.2.41](#) in [IBM Storage Scale FAQ](#).

Related concepts

[NFS V4 ACL Syntax](#)

[NFS V4 ACL translation](#)

[Considerations when using GPFS with NFS V4 ACLs](#)

Related tasks

[Displaying NFS V4 access control lists](#)

[Applying an existing NFS V4 access control list](#)

[Changing NFS V4 access control lists](#)

[Deleting NFS V4 access control lists](#)

Related reference

[Exceptions and limitations to NFS V4 ACLs support](#)

Review the exceptions and limitations to NFS V4 ACLs in IBM Storage Scale.

Displaying NFS V4 access control lists

The `mmgetacl` command displays an existing ACL regardless of its type (traditional or NFS V4). The format of the ACL that is returned depends on the file system setting (-k flag), and the format of the actual ACL associated with the file. For more information, see “[NFS V4 ACL translation](#)” on page 469.

Starting from IBM Storage Scale 5.1.7, IBM Storage Scale supports setting the extended **system.nfs4_acl** attribute as another method for manipulating NFSv4 ACLs. The **nfs4_getfac1** command can be used directly in IBM Storage Scale to get NFSv4 ACLs. For requirements and limitations, see [Q.2.41 in IBM Storage Scale FAQ](#).

Related concepts

[NFS V4 ACL Syntax](#)

[NFS V4 ACL translation](#)

[Considerations when using GPFS with NFS V4 ACLs](#)

Related tasks

[Setting NFS V4 access control lists](#)

[Applying an existing NFS V4 access control list](#)

[Changing NFS V4 access control lists](#)

[Deleting NFS V4 access control lists](#)

Related reference

[Exceptions and limitations to NFS V4 ACLs support](#)

Review the exceptions and limitations to NFS V4 ACLs in IBM Storage Scale.

Applying an existing NFS V4 access control list

This function is identical, whether using traditional or NFS V4 ACLs. See “[Applying an existing traditional GPFS access control list](#)” on page 464.

Related concepts

[NFS V4 ACL Syntax](#)

[NFS V4 ACL translation](#)

[Considerations when using GPFS with NFS V4 ACLs](#)

Related tasks

[Setting NFS V4 access control lists](#)

[Displaying NFS V4 access control lists](#)

[Changing NFS V4 access control lists](#)

[Deleting NFS V4 access control lists](#)

Related reference

[Exceptions and limitations to NFS V4 ACLs support](#)

Review the exceptions and limitations to NFS V4 ACLs in IBM Storage Scale.

Changing NFS V4 access control lists

This function is identical, whether using traditional or NFS V4 ACLs. For more information, see “[Changing traditional GPFS access control lists](#)” on page 465.

Related concepts

[NFS V4 ACL Syntax](#)

[NFS V4 ACL translation](#)

[Considerations when using GPFS with NFS V4 ACLs](#)

Related tasks

[Setting NFS V4 access control lists](#)

[Displaying NFS V4 access control lists](#)

[Applying an existing NFS V4 access control list](#)

[Deleting NFS V4 access control lists](#)

Related reference

[Exceptions and limitations to NFS V4 ACLs support](#)

Review the exceptions and limitations to NFS V4 ACLs in IBM Storage Scale.

Deleting NFS V4 access control lists

Use the `mmdelacl` command to delete NFS V4 ACLs. After the ACL is deleted, the permissions revert to the mode bits. If the `mmgetacl` command is used to display the ACL (`mmgetacl -k native`), it appears as a traditional GPFS ACL.

When assigning an ACL to a file that already has an NFS V4 ACL, there are some NFS rules that must be followed. Specifically, in the case of a directory, there will not be two separate (access and default) ACLs, as there are with traditional ACLs. NFS V4 needs a single ACL entity and allows individual ACL entries to be flagged if they are to be inherited. Therefore, `mmputacl -d` isn’t allowed if the existing ACL was the NFS V4 type, since this attempts to change only the default ACL. Likewise, `mmputacl` (without the `-d` flag) isn’t allowed because it attempts to change only the access ACL, leaving the default unchanged. To change such an ACL, use the `mmeditacl` command to change the entire ACL as a unit. You can also use the `mmdelacl` command to remove an NFS V4 ACL, followed by the `mmputacl` command.

Related concepts

[NFS V4 ACL Syntax](#)

[NFS V4 ACL translation](#)

[Considerations when using GPFS with NFS V4 ACLs](#)

Related tasks

[Setting NFS V4 access control lists](#)

[Displaying NFS V4 access control lists](#)

[Applying an existing NFS V4 access control list](#)

[Changing NFS V4 access control lists](#)

Related reference

[Exceptions and limitations to NFS V4 ACLs support](#)

Review the exceptions and limitations to NFS V4 ACLs in IBM Storage Scale.

Considerations when using GPFS with NFS V4 ACLs

There are several constraints that you need to consider when using GPFS with NFS V4 ACLs. For a comprehensive description of these restrictions, see “[Exceptions and limitations to NFS V4 ACLs support](#)” on page 473.

Related concepts

[NFS V4 ACL Syntax](#)

[NFS V4 ACL translation](#)

Related tasks

[Setting NFS V4 access control lists](#)

[Displaying NFS V4 access control lists](#)

[Applying an existing NFS V4 access control list](#)

[Changing NFS V4 access control lists](#)

[Deleting NFS V4 access control lists](#)

Related reference

[Exceptions and limitations to NFS V4 ACLs support](#)

Review the exceptions and limitations to NFS V4 ACLs in IBM Storage Scale.

Exceptions and limitations to NFS V4 ACLs support

Review the exceptions and limitations to NFS V4 ACLs in IBM Storage Scale.

1. IBM Storage Scale has limited support for ACLs, but only with Samba on Linux. In that environment, IBM Storage Scale can only save and retrieve Alarm and Audit access control entries (ACEs). No actions are defined that can be taken for ACEs during ACL evaluation.
2. Some types of access for which NFS V4 defines controls don't currently exist in IBM Storage Scale. For these, ACL entries are accepted and saved, but because there's no corresponding operation they have no effect. These include **READ_NAMED**, **WRITE_NAMED**, and **SYNCHRONIZE**.
Note: Even if IBM Storage Scale ignores these bits, the SMB service enforces them on the protocol level.
3. AIX requires that **READ_ACL** and **WRITE_ACL** always be granted to the object owner. Although granting these ACL entries contradicts NFS Version 4 protocol, it's considered as an area where users would otherwise erroneously leave an ACL that only privileged users can change. Since ACLs are file attributes, **READ_ATTR** and **WRITE_ATTR** are similarly granted to the owner. As it wouldn't make sense to then prevent the owner from accessing the ACL from a non-AIX node, IBM Storage Scale has implemented this exception everywhere.
4. AIX does not support the use of special name values other than **owner@**, **group@**, and **everyone@**. Therefore, these are the only valid special name values for use in IBM Storage Scale NFS V4 ACLs.
5. NFS V4 allows ACL entries that grant permission to users or groups to change the ownership of a file with a command such as the **chown** command. For security reasons, IBM Storage Scale now restricts these permissions so that a non-privileged user can **chown** such a file only to self or to a group that the user is a member of.
6. With some limitations, Windows clients that access IBM Storage Scale through Samba can use their native NTFS ACLs, which are mapped to the underlying NFS v4 ACLs. For limitations, see [“Authorization limitations” on page 496](#).
7. Ganesha supports NFS v4 ACLs to and from IBM Storage Scale. However, to export a file system with cNFS/KNFS, you must configure the file system to support POSIX ACLs. Use the **mmcrfs** command with the **-k all** or **-k posix** parameter. With Samba, use the **-k nfs4** parameter. NFS V4 Linux servers handle ACLs properly only if they're stored in GPFS as POSIX ACLs. For more information, see [“Linux ACLs and extended attributes” on page 474](#).
8. The cluster can include Samba, CES NFS, AIX NFS, and IBM Storage Scale Windows nodes.
9. NFS V4 ACLs can be stored in GPFS file systems using Samba exports, NFS V4 AIX servers, GPFS Windows nodes, **ac1put**, and **mmputac1**. Clients of Linux V4 servers can't see stored ACLs but can see the permissions from the mode.
10. Starting from IBM Storage Scale 5.1.7, IBM Storage Scale supports setting the extended **system.nfs4_ac1** attribute as another method for manipulating NFSv4 ACLs. You can employ the syntax of these tools to manage NFSv4 ACLs in IBM Storage Scale. For requirements and limitations, see [Q.2.41 in IBM Storage Scale FAQ](#).

For more information about ACLs and NFS export, see [Chapter 36, “Managing GPFS access control lists,” on page 461](#).

Related concepts

[NFS V4 ACL Syntax](#)

[NFS V4 ACL translation](#)

[Considerations when using GPFS with NFS V4 ACLs](#)

Related tasks

[Setting NFS V4 access control lists](#)

[Displaying NFS V4 access control lists](#)

[Applying an existing NFS V4 access control list](#)

[Changing NFS V4 access control lists](#)

[Deleting NFS V4 access control lists](#)

Linux ACLs and extended attributes

NFS V4 uses the existing POSIX ACLs and the extended attribute support in Linux that is supported by GPFS.

Note: This topic applies only to cNFS/kNFS.

Although the NFS V4 protocol defines a richer ACL model similar to Windows ACLs, the Linux implementation maps those ACLs to POSIX ACLs before passing them to the underlying file system. This mapping is done in nfsd indicating that on setting an ACL from a client and then fetching it, the server does not return exactly what you have set. This discrepancy is because the ACL you set was converted to a POSIX ACL and then back again.

NFS V4 ACLs are more fine-grained than POSIX ACLs, so the POSIX-to-NFS V4 translation is close to perfect, but the NFS V4-to-POSIX translation isn't. The NFS V4 server tries to err on the side of mapping to a stricter ACL. There's a small set of NFS V4 ACLs that the server rejects completely (such as, any ACL that attempts to explicitly DENY permission to read attributes). Except for these ACLs, the server tries hard to accept nearly all other ACLs and map them as best it can.

ACLs that are set through AIX/NFS V4 and Windows nodes are written as NFS V4 ACLs to GPFS. ACLs that are set through Linux/NFS V4 are written as POSIX ACLs to GPFS. Currently, GPFS does not provide an interface to convert on-disk NFS V4 ACLs to POSIX ACLs. This means that if ACLs are written through either AIX/NFS V4 or Windows, they cannot be read by Linux/NFS V4. In this case, a Linux NFS V4 server constructs an ACL from the permission mode bits only and ignores the ACL on the file.

Authorizing protocol users

Authorization grants or denies access to resources such as directories, files, commands, and functions. Authorization is applicable to an already authenticated identity, such as an IBM Storage Scale data user, an administrative user, or an IBM service representative. Access to the files and directories of the IBM Storage Scale system is managed through Access Control Lists (ACLs). It ensures that only authorized users get access to exports, directories, and files. An Access Control Entry (ACE) is an individual entry in an access control list, and describes the permissions for an individual user or group of users. An ACL can have zero or more ACEs.

Related concepts

[Traditional GPFS ACL administration](#)

Support for NFS V4 access control lists (ACLs) is added to traditional ACL support. NFS V4 ACLs are different than the traditional ones.

[NFS V4 ACL administration](#)

AIX does not allow a file system to be NFS V4 exported unless it supports NFS V4 ACLs. By contrast, Linux does not allow a file system to be NFS V4 exported unless it supports POSIX ACLs.

Authorizing file protocol users

The IBM Storage Scale system uses ACLs to authorize users who access the system through file protocols such as NFS and SMB.

The GPFS file system supports storing POSIX and NFSv4 ACLs to authorize file protocol users.

The SMB service maps the NFSv4 ACL to a Security Descriptor for SMB clients to form the ACLs. Any ACL changes from SMB clients are mapped back to the NFSv4 in the file system and stored there. There is only the NFSv4 ACL stored in the file system.

CES protocol services are only supported with file systems using NFSv4 ACLs. To get the expected behavior of ACLs, you must configure the file system to use only NFSv4 ACLs. The default configuration profiles (`/usr/lpp/mmfs/profiles`) that are included with IBM Storage Scale contain the required configuration for NFSv4 ACLs in the file system. When creating a file system for protocol usage with the **mmcrfs** command without using profiles, the `-k nfs4` option needs to be added. For more information, see the *mmcrfs command* and the *mmchfs command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

ACLs can be applied at the following levels:

- Files and directories
- SMB shares

Note: Note that ACLs from the object protocol are not mapped to the file system ACLs and are stored separately.

ACLs on files and directories

With the SMB and NFS protocols, you can manage the ACL permissions on files and directories from connected file systems. ACLs from both protocols (NFS and SMB) are mapped to the same ACL in the file system. The ACL supports inheritance and you can control the inheritance by using the special inheritance flags.

It is a good practice to manage ACLs from the ACL management interface on a SMB or NFS client system. For example, after you create the directories for an SMB or NFS export, you can set the owner of the directory with the **chown** command. Then the user can connect to the export with the SMB or NFS protocol to see and manage the ACLs that are associated with the directory over which the export is created.

Setting ACLs from protocol clients allows using ACLs in exactly the format expected by the client. Especially Microsoft Windows Clients expect the ACL to be stored in a canonical order. To avoid problems with presenting differently ordered ACLs to these clients, manage the ACLs for SMB clients that run Microsoft Windows from a Microsoft Windows system.

ACLs and POSIX mode bits

The POSIX bits of a file or directory are another authorization method, different from ACLs. POSIX bits can also be used to specify access permissions for a file. You can use the POSIX bits of a file to configure access control for an owner, a group, and for all users to read, update, or run the file. POSIX bits are less flexible than ACLs.

As the file system only stores the NFSv4 ACL, changing the POSIX mode bits also modifies the ACL of an object. When you use ACLs for access control, consider how changes to POSIX mode bits should interact with the ACL. You can configure this behavior with the `--allow-permission-change` parameter of the **mmcrfileset** and **mmchfileset** commands.

An ACL extends the base permissions or the standard file access modes such as read, write, and execute. ACLs are compatible with UNIX mode bits. Issuing the `chmod` command by the NFS clients overwrite

the access privileges that are defined in the ACL by the privileges that are derived from UNIX mode bits. By default, the ACLs are replaced by UNIX mode bits if the chmod command is submitted. To allow proper use of ACLs, it is a good practice to prevent chmod from overwriting the ACLs by setting the `--allow-permission-change` parameter of the `mmcrfileset` or `mmchfileset` command parameter to `setAclOnly` or `cchmodAndUpdateAcl`.

This is especially important for applications running on cluster nodes and for NFSv3 clients, as those will issue chmod calls to set the POSIX mode bits. These will be stored in the ACL, depending on the `--allow-permission-change` parameter.

The permissions from the NFSv4 ACL entries `special:owner@` are shown as the POSIX permission bits for the file owner, `special:group@` are shown as the POSIX permission bits for the group, and `special:everyone@` are shown for the POSIX permissions for “other”.

SMB protocol share-level ACLs

SMB share ACLs apply only to SMB exports and they are separate from the file system ACLs. The default for share-level ACLs is granting full access to everybody, resulting only in the file system ACL determining access by default. When SMB share ACLs are set to restrict access, users accessing data through the SMB protocol need to have access in both, the share-level ACL and in the file system ACL.

SMB share ACLs can be changed either through the MMC on a Windows client or through the `mmsmb exportacl` command. For more information, see the `mmsmb exportacl` command on any protocol node.

Mapping between SMB protocol Security Descriptors and NFSv4 ACLs stored in the file system

The SMB protocol uses Security Descriptors to describe the permissions on a file or directory. The Discretionary Access Control Lists (DACL) from the Security Descriptors are mapped to and from the NFSv4 ACLs stored in the file system. That way, only one authoritative ACL exists for each file or directory.

The structure of NFSv4 ACLs and DACLs is similar. The ACL consists of a list of entries. Each entry contains the elements: The principal (the user or group the entry is referring to), inheritance flags, the type (allow or deny) and the permissions being granted or denied in this entry.

Depending on the configured id mapping method for a domain, the mapping from the Security Descriptor to the NFSv4 ACL is also done slightly differently. The reason here is that the `-unixmap-domain` and `-ldapmap-domain` methods strictly map the SID of an Active Directory user to a uid and the SID of an Active Directory group to a gid. The default internal mapping method that is used when no other method has been specified maps each Active Directory user or group to both, a uid and a gid with the same numeric value. That allows for file ownership of a group to be represented in the IBM Storage Scale file system.

When possible, the Security Descriptor entries matching owner or owning group are mapped to the NFSv4 ACL `special:owner@` and `special:group@` entries, so that the permissions are also reflected in the POSIX mode bits. This is not possible when the Security Descriptor entries have flags to be inherited to new files or folders, as inheriting the entries to new files or folders with different owner information would create unwanted entries.

Multiple ACL entries which are exact duplicates after the mapping is done are merged into one ACL entry.

The mapping from the NFSv4 ACL to an SMB Security Descriptor is done according to the following table:

Table 35. Mapping from NFSv4 ACL entry to SMB Security Descriptor				
Entry in NFSv4 ACL		Mapped to SMB Security Descriptor entry		
Principal	Inheritance	Principal	Inheritance	Comment
special:everyone@	Any	EVERYONE	Same	NA

Table 35. Mapping from NFSv4 ACL entry to SMB Security Descriptor (continued)

Entry in NFSv4 ACL		Mapped to SMB Security Descriptor entry		
Principal	Inheritance	Principal	Inheritance	Comment
special:owner@	Includes FileInherit or DirInherit	CREATOR OWNER	Same with the exclusion of the current file or folder added	An entry that applies to the current file or folder and marked for inheritance can result in mapping one NFS4 entry in two Security Descriptor entries.
	Applies to current folder (not InheritOnly)	Mapped user	Same	
special:group@	Includes FileInherit or DirInherit	CREATOR GROUP	Same with the exclusion of the current file or folder added	An entry that applies to the current file or folder and marked for inheritance can result in mapping one NFS4 entry in two Security Descriptor entries.
	Applies to current folder (not InheritOnly)	Mapped group	Same	
Explicit entry	Any	Matching principal	Same	NA

Mapping between SMB protocol Security Descriptors and NFSv4 ACL with the --unixmap-domains or --ldapmap-domains id mappings

In this case, each user in Active Directory is mapped to a uid and each group in Active Directory is mapped to a gid. This carries the limitation that a file or directory cannot be owned by a group as the owner of an object in the IBM Storage Scale file system is always a uid, not a gid. The mapping of entries from an SMB Security Descriptor to a NFSv4 ACL in this case is done according to the following table:

Table 36. Mapping from SMB Security Descriptor to NFSv4 ACL entry with unixmap or ldapmap id mapping

Entry in SMB Security Descriptor		Mapped to NFSv4 ACL entry	
Principal	Inheritance	Principal	Inheritance
EVERYONE	Any	special:everyone@	Same
CREATOR OWNER	Subfolders or files	special:owner@	Same with InheritOnly added
CREATOR GROUP	Subfolders or files	special:group@	Same with InheritOnly added

Table 36. Mapping from SMB Security Descriptor to NFSv4 ACL entry with unixmap or ldapmap id mapping (continued)

Entry in SMB Security Descriptor		Mapped to NFSv4 ACL entry	
Principal	Inheritance	Principal	Inheritance
(Principal matching owner of file or directory)	This owner only	special:owner@ Exception: DENY entries for the owner which deny attribute or ACL access are mapped to named entry instead, due to limitation in file system.	None
	Any inheritance to subfolders or files	Named entry for owning user, not special:owner@	Same
(Principal matching owning group of file or directory)	This folder only	special:group@	None
	Any inheritance to subfolders or files	Named entry for owning group, not special:group@	Same
(Other principal)	Any	Named entry	Same

Mapping between SMB protocol Security Descriptors and NFSv4 ACL with the internal id mapping

The default id mapping method creates id mappings based on an internal database on the CES protocol nodes. This id mapping method assigns both, a uid and a gid, to each SID. This has the advantage that files and objects can be owned by a group. This also affects the mapping from the Security Descriptor to the NFSv4 ACL, as most entries are now mapped to gid entries:

Table 37. Mapping from SMB Security Descriptor to NFSv4 ACL entry with default id mapping

Entry in SMB Security Descriptor		Mapped to NFSv4 ACL entry	
Principal	Inheritance	Principal	Inheritance
EVERYONE	Any	special:everyone@	Same
CREATOR OWNER	Subfolders or files	special:owner@	Same with InheritOnly added
CREATOR GROUP	Subfolders or files	special:group@	Same with InheritOnly added

Table 37. Mapping from SMB Security Descriptor to NFSv4 ACL entry with default id mapping (continued)

Entry in SMB Security Descriptor		Mapped to NFSv4 ACL entry	
Principal	Inheritance	Principal	Inheritance
(Principal matching owner of file or directory)	This folder only	Named group entry representing user and additional special:owner@ entry to reflect permissions in POSIX mode bits Exception: DENY entries for the owner with deny attribute or ACL access are written with a named user entry instead of the special:owner@ entry, due to limitation in file system.	None
	Any inheritance to subfolders, filesubfolders, or files	Named group entry for owning user	Same
(Principal matching owning group of file or directory)	None. This folder only	special:group@	None
	Any inheritance to subfolders or files	Named entry for owning group, not special:group@	Same
(Other principal)	Any	Named entry	Same

Related concepts

[Authorizing object users](#)

The Object Storage service of the IBM Storage Scale system uses Keystone service for identity management. The identity management service consists of user authentication and authorization processes.

[Authorization limitations](#)

Authorization limitations are specific to the protocols that are used to access data.

ACL inheritance

The inheritance flags in ACL entry of parent directories are used to control the inheritance of authorization to the child files and directories. The inheritance flag gives you the granularity to specify whether the inheritance defined in an ACL entry applies to the current directory and its children or only to the subdirectories and files that are contained in the parent directory. ACL entries are inherited to the child directories or files at the time of creation. Changes made to the ACL of a parent directory are not propagated to child directories or files. However, in case of SMB, you can specify to propagate the inheritance changes from a parent to all its child by using File Explorer, command line, or PowerShell.

Controlling inheritance of entries inside an ACL

The NFSv4 protocol uses the following flags to specify and control inheritance information of the ACEs:

- *FileInherit*: Indicates that this ACE must be added to each new non-directory file created. This flag is signified by 'f' or file_inherit.

- *DirInherit*: Indicates that this ACE must be added to each new directory created. This flag is signified by ‘d’ or `dir_inherit`.
- *InheritOnly*: Indicates that this ACE is not applied to the parent directory itself, but only inherited by its children. This flag is signified by ‘i’ or `inherit_only`.
- *NoPropagateInherit*: Indicates that the ACL entry must be included in the initial ACL for subdirectories that are created in this directory but not further propagated to subdirectories created below that level.

In case of SMB, the following list shows how the inheritance flags are linked to the Microsoft Windows inheritance modes:

- This folder only (No bits)
- This folder, subfolder, and files (FileInherit, DirInherit)
- This folder and subfolders (DirInherit)
- This folder and files (FileInherit)
- Subfolders and files only (FileInherit, DirInherit, InheritOnly)
- Subfolders only (DirInherit, InheritOnly)
- Files only (FileInherit, InheritOnly)

Related concepts

ACL best practices

It is essential to properly apply ACLs to the file systems, filesets, and exports, and directories and files to ensure smooth access for the users.

ACL permissions that are required to work on files and directories

The topic describes the required ACL permissions to access files and folders through file protocols.

Working with ACLs

The IBM Storage Scale system applies default ACLs for newly created IBM Storage Scale file system components such as file system, filesets, file, directories, and exports.

Related tasks

Configuring file system ACL by using GUI

Managing ACLs works only when the file system supports NFSv4 ACL semantics. If the file system supports both POSIX and NFSv4 ACLs, the GUI reads the POSIX ACL in a compatible mode and while modifying the ACL, it overwrites POSIX ACLs with NFSv4 ACLs. When only POSIX is supported, writing ACLs through the GUI fails. You can manually enter the ACL values or define a template to load the standard values that are defined based on your requirement.

ACL best practices

It is essential to properly apply ACLs to the file systems, filesets, and exports, and directories and files to ensure smooth access for the users.

Consider the following points before you create or copy data into the export:

1. Should a group of users be given permission to access the data?
2. Should individual users be given permission to access the data?
3. Should selected users from different groups be given access to selected data?
4. Should the shares be in mixed mode? That is, do you have NFS and SMB clients who access the exports in explicit mode, where the data is accessed either from SMB or NFS?
5. Should the applications that the clients are using over SMB and NFS be given any specific ACL permission?

Setting ACLs for groups

The recommended way to manage access is per group instead of per individual user. This way, users can be easily added to or removed from the group. Providing ACLs to groups has an added advantage of managing inheritance easily for the whole group of users simultaneously. If individual users are added

directly to ACLs and you need to make a change, you need to update ACLs of all corresponding directories and files. On the authentication server like Active Directory or LDAP, you can create groups and add users as members and use these groups to give respective access to data.

Setting ACLs for individual users

If you need to set ACLs for individual users where data is created by users in folders that are created by others, it is recommended that you explicitly add the users who need ACLs on that export.

In mixed mode, where the share is used for both NFS and SMB access, parent owner might experience loss of access to the child directory or the files. To avoid such a problem, it is recommended that you provide ACLs explicitly to each user.

Special Owner and Group

The special owner and group dynamically refer to the owner and group of the directory or file that the ACL belongs to. For example, if the owner of a file is changed, all special:owner@ entries in the ACL refers to the new owner. In case of inheritance, this leads to some complexity because those special entries point to the owner and group of the child directory or file that inherits the entry. In many cases, the children do not have the same owner and group as the parent directory. Therefore, the special entries in parent and children refer to different users. This can be avoided by adding static entries (user:'name' or group:'name') to the ACL. These static entries are inherited by name and refer everywhere to the same users. But they are not updated if the owner of the parent is changed. The general recommendation is not to use special:owner@ and special:group@ together with inheritance flags. For more information, see the **mmputacl** command.

Setting ACLs for special IDs

The inheritance of ACL from the owner of a directory to subdirectories and files works only for subdirectories and files that have the same owner as the parent directory. A subdirectory or file that is created by a different owner does not inherit the ACL of a parent directory that is owned by another user.

In case of special access to NFSV4 exports, parent owners might experience loss of access to its child folders and files. To avoid such a problem, for mixed mode, it is recommended that you provide ACLs to groups rather than to individual users.

Related concepts

ACL inheritance

The inheritance flags in ACL entry of parent directories are used to control the inheritance of authorization to the child files and directories. The inheritance flag gives you the granularity to specify whether the inheritance defined in an ACL entry applies to the current directory and its children or only to the subdirectories and files that are contained in the parent directory. ACL entries are inherited to the child directories or files at the time of creation. Changes made to the ACL of a parent directory are not propagated to child directories or files. However, in case of SMB, you can specify to propagate the inheritance changes from a parent to all its child by using File Explorer, command line, or PowerShell.

ACL permissions that are required to work on files and directories

The topic describes the required ACL permissions to access files and folders through file protocols.

Working with ACLs

The IBM Storage Scale system applies default ACLs for newly created IBM Storage Scale file system components such as file system, filesets, file, directories, and exports.

Related tasks

Configuring file system ACL by using GUI

Managing ACLs works only when the file system supports NFSV4 ACL semantics. If the file system supports both POSIX and NFSv4 ACLs, the GUI reads the POSIX ACL in a compatible mode and while modifying the ACL, it overwrites POSIX ACLs with NFSV4 ACLs. When only POSIX is supported, writing ACLs through the GUI fails. You can manually enter the ACL values or define a template to load the standard values that are defined based on your requirement.

ACL permissions that are required to work on files and directories

The topic describes the required ACL permissions to access files and folders through file protocols.

The following table describes the ACL permissions that are required when the user of the file is not the file owner, where "X" denotes permission that is required on file or directory and "P" denotes permission that is required on the parent directory of the file or directory.

Note: In IBM Storage Scale 5.0.3, a difference in the handling of the NFSv4 ACL bit SYNCHRONIZE can cause access issues for Microsoft Windows clients. The change is that when ACL data is returned to the SMB client, the SYNCHRONIZE bit on ACL "allow" entries is passed unchanged. But Microsoft Windows clients require the SYNCHRONIZE bit to be set for renaming files or directories. Files that are written by Microsoft Windows clients usually have the SYNCHRONIZE bit set.

To restore the pre-5.0.3 behavior, issue the following command for each SMB share that is affected by the problem:

```
/usr/lpp/mmf/bin/net conf setparm <SMBShareName> 'nfs4:set synchronize' yes
```

In the long term, it is a good idea to change the ACLs for all files and directories that are missing the SYNCHRONIZE bit instead of modifying the SMB configuration.

Table 38. ACL permissions required to work on files and directories, while using SMB protocol (table 1 of 2)

ACL Operation	ACL Permission					
	Traverse folder / execute file	List folder / read data	Read attribute	Read extended attribute	Create files / write data	Create folders / append data
Execute file	X	X				
List folder		X				
Read data from file		X	X	X		
Read attributes			X			
Create file					X	
Create folder						X
Write data to file		X	X		X	X
Write file attributes						
Write folder attributes						
Delete file		P	X		P	
Delete folder		P	X		P	
Rename file		P	X		P	
Rename folder		P	X		P	P
Read file permissions						
Read folder permissions						
Write file permissions						
Write folder permissions						
Take file ownership						

*Table 38. ACL permissions required to work on files and directories, while using SMB protocol (table 1 of 2)
(continued)*

ACL Operation	ACL Permission					
	Traverse folder / execute file	List folder / read data	Read attribute	Read extended attribute	Create files / write data	Create folders / append data
Take folder ownership						

Table 39. ACL permissions required to work on files and directories, while using SMB protocol (table 2 of 2)

ACL Operation	ACL Permission						
	Write attribute	Write extended attributes	Delete subfolder and files	Delete	Read permission s	Write permission s	Take ownership
Execute file							
List folder							
Read data from file							
Read attributes							
Create file							
Create folder							
Write data to file	X	X					
Write file attributes	X						
Write folder attributes	X						
Delete file			P or X				
Delete folder			P or X				
Rename file			P or X				
Rename folder			P or X				
Read file permissions					X		
Read folder permissions					X		
Write file permissions					X	X	
Write folder permissions					X	X	
Take file ownership							X
Take folder ownership							X

Table 40. ACL permissions required to work on files and directories, while using NFS protocol (table 1 of 2)

ACL Operation	ACL Permission					
	Traverse folder / execute file	List folder / read data	Read attribute	Read extended attribute	Create files / write data	Create folders / append data
Execute file	P, X	X				
List folder	P	X				
Read data from file	P	X				
Read attributes	P					
Create file	P				P	
Create folder	P					P
Write data to file	P				X	X
Write file attributes	P					
Write folder attributes	P					
Delete file	P				P	
Delete folder	P				P	
Rename file	P		X		P	
Rename folder	P		X		P	P
Read file ACL	P					
Read folder ACL	P					
Write file ACL	P					
Write folder ACL	P					
Take file ownership	P					
Take folder ownership	P					

Table 41. ACL permissions required to work on files and directories, while using NFS protocol (table 2 of 2)

ACL Operation	ACL Permission						
	Write attribute	Write extended attributes	Delete subfolder and files	Delete	Read ACL	Write ACL	Take ownership
Execute file							
List folder							
Read data from file							
Read attributes							
Create file							
Create folder							
Write data to file							

Table 41. ACL permissions required to work on files and directories, while using NFS protocol (table 2 of 2)
(continued)

ACL Operation	ACL Permission						
	Write attribute	Write extended attributes	Delete subfolder and files	Delete	Read ACL	Write ACL	Take ownership
Write file attributes							
Write folder attributes							
Delete file			P				
Delete folder			P				
Rename file			P				
Rename folder			P				
Read file ACL							
Read folder ACL							
Write file ACL						X	
Write folder ACL						X	
Take file ownership							X
Take folder ownership							X

The following are the considerations on the ACL read and write permissions:

1. The files that require "Traverse folder / execute file" permission do not require the "Bypass Traverse Check" attribute to be enabled. This attribute is enabled by default on the files.
2. The "Read extended attribute" permission is required by the SMB clients with recent Microsoft Windows versions (for Microsoft Windows 2008, Microsoft Windows 2012, and Microsoft Windows 8 versions) for file copy operations. The default ACLs set without inheritance do not contain this permission. It is recommended that you use inherited permissions where possible and enable this permission in the inherited permissions to prevent the default value to be used and cause problems.

Migrating data through SMB to the IBM Storage Scale cluster requires a user ID with the enhanced permissions. The ownership of a file cannot be migrated by a normal IBM Storage Scale user. Therefore, you need to configure an “admin user” to allow data migration. For more information on how to configure the “admin users” parameter, see the *mmsmb export add* and *mmsmb export change* sections in *mmsmb command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Directory traversal permissions that are applicable for SMB ACLs

The following are the considerations on the traverse permissions:

1. It is recommended that you add the "Traverse folder / execute file" permission to all executable files, even if the "Bypass Traverse Check" attribute is enabled on these files. IBM Storage Scale checks for the "Traverse folder / execute file" permission on executable files irrespective of the value of the "Bypass Traverse Check" attribute.
2. If the `--cifsBypassTraversalChecking` option is enabled, it allows a user to directly access files and folders that the user owns, and also that are contained under the parent folders for which the user does not have Read or Write permissions. Users without "Read and Execute" access to the share or export in which the user-owned files and folders are located can read and modify the files inside the export for which the user has permissions that are granted by the

--cifsBypassTraversalChecking option. However, in this case, operations like rename file and delete file are not granted by default. This is normal SMB behavior. Modify ACLs as required to enable these operations.

For example, in the directory structure /A/B/C, assume that an SMB user has 'read' permission on C but no permissions on A and B. When the --cifsBypassTraversalChecking option is set to its default value Yes, this SMB user can access C without having "Traverse Folder" or "Execute File" permissions that are set to allow on A and B, but is still not allowed to browse the content of A and B.

3. The ownership of a file cannot be migrated by a normal user. You must configure and use administrative user credentials to perform data migration. When migrating existing files and directories from other systems to IBM Storage Scale, the ACL might not contain explicit traversal rights for the users because the source system can grant this right implicitly. After migrating the files with ACLs, ensure that traversal rights are granted to the parent directory of each exported path.

Related concepts

ACL inheritance

The inheritance flags in ACL entry of parent directories are used to control the inheritance of authorization to the child files and directories. The inheritance flag gives you the granularity to specify whether the inheritance defined in an ACL entry applies to the current directory and its children or only to the subdirectories and files that are contained in the parent directory. ACL entries are inherited to the child directories or files at the time of creation. Changes made to the ACL of a parent directory are not propagated to child directories or files. However, in case of SMB, you can specify to propagate the inheritance changes from a parent to all its child by using File Explorer, command line, or PowerShell.

ACL best practices

It is essential to properly apply ACLs to the file systems, filesets, and exports, and directories and files to ensure smooth access for the users.

Working with ACLs

The IBM Storage Scale system applies default ACLs for newly created IBM Storage Scale file system components such as file system, filesets, file, directories, and exports.

Related tasks

Configuring file system ACL by using GUI

Managing ACLs works only when the file system supports NFSv4 ACL semantics. If the file system supports both POSIX and NFSv4 ACLs, the GUI reads the POSIX ACL in a compatible mode and while modifying the ACL, it overwrites POSIX ACLs with NFSv4 ACLs. When only POSIX is supported, writing ACLs through the GUI fails. You can manually enter the ACL values or define a template to load the standard values that are defined based on your requirement.

Working with ACLs

The IBM Storage Scale system applies default ACLs for newly created IBM Storage Scale file system components such as file system, filesets, file, directories, and exports.

The file system must be created with native ACL type as NFS V4. It is recommended that you use the default configuration profiles (/usr/lpp/mmfs/profiles) that are included with IBM Storage Scale. It contains the required configuration for NFSv4 ACLs in the file system.

Applying default ACLs

Perform the following steps to apply default ACLs on SMB and NFS exports:

1. Create a fileset or directory in the file system as shown in the following example:

```
mkdir -p /ibm/gpfs0/testsmlexport
```

2. Change the owner and group of the fileset or directory using **chown** and **chgrp** respectively. For example:

```
chown -R "DOMAIN\\username": "DOMAIN\\groupname" /ibm/gpfs0/testsmlexport
```

3. Use the **mmputacl** or **mmeditacl** commands to set the wanted ACE along with specific ACE for owner user and owner group and inheritance flags for the fileset or directory.
4. Check the ACL setting for the fileset or directory by using the **mmgetacl** command.
5. Create the desired SMB or NFS export by using the **mmnfs** or **mmsmb** commands over the fileset or directory.
6. For data exported for SMB clients, it is recommended that you manage the ACLs from a Windows clients, since there is already a GUI interface available and the ACL is set according to the requirements of Windows clients. Modifying the ACL directly with **mmputacl** and **mmeditacl** are not advised.

Viewing the owner of the SMB share

Perform the following steps to create an SMB share and view the owner of the export:

1. Submit the **mmsmb export add** command to create an SMB share as shown in the following example:

```
mmsmb export add testsmbexport /ibm/gpfs0/testsmbexport
```

2. Issue either the **ls -la** command or the **mmgetacl** command to view the owner of the export. For example:

```
ls -la /ibm/gpfs0/testsmbexport
```

Or

```
mmgetacl /ibm/gpfs0/testsmbexport
```

Apart from the tasks that are listed earlier in this section, the following table provides a quick overview of the tasks that can be performed to manage ACLs and the corresponding IBM Storage Scale command.

<i>Tasks that can be performed to manage ACLs</i>	<i>Command</i>	<i>Reference topic</i>
Applying ACL at file system, fileset, and export level	mmeditacl	“Applying an existing NFS V4 access control list” on page 471
Inserting ACEs in existing ACLs	mmeditacl	“Changing NFS V4 access control lists” on page 472
Modifying ACLs	mmeditacl	“Changing NFS V4 access control lists” on page 472
Copying Access control list entries	mmeditacl	“Changing NFS V4 access control lists” on page 472
Replacing a complete ACL	mmputacl or mmeditacl	“Changing NFS V4 access control lists” on page 472
Replacing all entries for a specific user inside an ACL	mmeditacl	“Changing NFS V4 access control lists” on page 472
Controlling inheritance of entries inside an ACL	mmputacl or mmeditacl	
Deleting complete ACL	mmdelacl	“Deleting NFS V4 access control lists” on page 472
Deleting specific ACL entries	mmeditacl	“Changing NFS V4 access control lists” on page 472

Table 42. Commands and reference to manage ACL tasks (continued)

Tasks that can be performed to manage ACLs	Command	Reference topic
Deleting ACL entry for a user	mmeditacl	“Changing NFS V4 access control lists” on page 472
Displaying an ACL	mmgetacl	“Displaying NFS V4 access control lists” on page 471
Changing file system directory’s owner and group	chown or chgroup	
Displaying file system directory’s owner and group	ls -l or mmgetacl	

Important: The **mmgetacl**, **mmputacl**, and **mmeditacl** commands are available to change the ACLs directly. As the SMB clients might depend on the order of entries in the ACL, it is not recommended that you change the ACLs directly on GPFS while using the SMB protocol. Changing an ACL directly in GPFS also does not account for inherited entries. So, it is recommended that you change the ACLs from a windows client.

Managing ACLs from Windows clients

For SMB shares, it is recommended that you manage the ACLs from a Windows client. The following operations are included in creating an SMB share:

1. Create the folder to export in the file system with the `mkdir` command.
2. Change the owner of the exported folder to a user who configures the initial ACLs.
3. Create the export using the `mmsmb export add` command.
4. Using a Windows client machine, access the newly created share as the user specified in step 2.
5. Right-click on the shared folder, and select **Properties**.
6. Select the **Security** tab and then select **Advanced** to navigate to the more detailed view of permissions.
7. Add and remove permissions as required.

Related concepts

ACL inheritance

The inheritance flags in ACL entry of parent directories are used to control the inheritance of authorization to the child files and directories. The inheritance flag gives you the granularity to specify whether the inheritance defined in an ACL entry applies to the current directory and its children or only to the subdirectories and files that are contained in the parent directory. ACL entries are inherited to the child directories or files at the time of creation. Changes made to the ACL of a parent directory are not propagated to child directories or files. However, in case of SMB, you can specify to propagate the inheritance changes from a parent to all its child by using File Explorer, command line, or PowerShell.

ACL best practices

It is essential to properly apply ACLs to the file systems, filesets, and exports, and directories and files to ensure smooth access for the users.

ACL permissions that are required to work on files and directories

The topic describes the required ACL permissions to access files and folders through file protocols.

Related tasks

Configuring file system ACL by using GUI

Managing ACLs works only when the file system supports NFSV4 ACL semantics. If the file system supports both POSIX and NFSV4 ACLs, the GUI reads the POSIX ACL in a compatible mode and while modifying the ACL, it overwrites POSIX ACLs with NFSV4 ACLs. When only POSIX is supported, writing

ACLs through the GUI fails. You can manually enter the ACL values or define a template to load the standard values that are defined based on your requirement.

Configuring file system ACL by using GUI

Managing ACLs works only when the file system supports NFSV4 ACL semantics. If the file system supports both POSIX and NFSv4 ACLs, the GUI reads the POSIX ACL in a compatible mode and while modifying the ACL, it overwrites POSIX ACLs with NFSV4 ACLs. When only POSIX is supported, writing ACLs through the GUI fails. You can manually enter the ACL values or define a template to load the standard values that are defined based on your requirement.

Note: Only GUI users with role *DataAccess* can configure or modify the file system ACL.

Perform the following steps to configure file system ACL:

1. Go to **Files > File System ACL** page in the IBM Storage Scale GUI.

You can either set an ACL based on an ACL template or create an ACL template from the **ACL Templates** tab.

2. Select the directory or folders for which you need to define the ACL. If you want to define ACL for the files, clear the **Only directories** checkbox so that files under the directories also become available for selection.
3. In the **Owner** and **Owning group** fields, type the owner and owning group of the path of the files or directory for which the ACL is being defined.
4. Define the ACL in the ACL section. For more information on the access permissions, click **Help Topic: Access Control List** option, which is available on the File System ACL page.
5. Instead of entering the details, you can load the access controls that are defined in an ACL template by using the **Load ACL Template** option.
6. Click **Edit** if you want to edit the already loaded ACL.
7. Click **Apply** to apply the new ACL to the files or the directory that is selected in the Directory field.

Related concepts

ACL inheritance

The inheritance flags in ACL entry of parent directories are used to control the inheritance of authorization to the child files and directories. The inheritance flag gives you the granularity to specify whether the inheritance defined in an ACL entry applies to the current directory and its children or only to the subdirectories and files that are contained in the parent directory. ACL entries are inherited to the child directories or files at the time of creation. Changes made to the ACL of a parent directory are not propagated to child directories or files. However, in case of SMB, you can specify to propagate the inheritance changes from a parent to all its child by using File Explorer, command line, or PowerShell.

ACL best practices

It is essential to properly apply ACLs to the file systems, filesets, and exports, and directories and files to ensure smooth access for the users.

ACL permissions that are required to work on files and directories

The topic describes the required ACL permissions to access files and folders through file protocols.

Working with ACLs

The IBM Storage Scale system applies default ACLs for newly created IBM Storage Scale file system components such as file system, filesets, file, directories, and exports.

Authorizing object users

The Object Storage service of the IBM Storage Scale system uses Keystone service for identity management. The identity management service consists of user authentication and authorization processes.

Important:

- CES Swift Object protocol feature is not supported from IBM Storage Scale 5.2.0 onwards.

- IBM Storage Scale 5.1.8 is the last release that has CES Swift Object protocol.
- IBM Storage Scale 5.2.0 will tolerate the update of a CES node from IBM Storage Scale 5.1.8.
 - *Tolerate* means:
 - The CES node will be updated to 5.2.0.
 - Swift Object support will not be updated as part of the 5.2.0 update.
 - You may continue to use the version of Swift Object protocol that was provided in IBM Storage Scale 5.1.8 on the CES 5.2.0 node.
 - IBM will provide usage and known defect support for the version of Swift Object that was provided in IBM Storage Scale 5.1.8 until you migrate to a supported object solution that IBM Storage Scale provides.
 - CES Swift Object is replaced with IBM Storage Scale S3. For more details, refer the *S3 support overview* section the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
 - For more information about Swift Object in IBM Storage Scale, refer to the [IBM Storage Scale 5.2.0 documentation](#).
- Please contact IBM for further details and migration planning.

Access for the object users to the Object Storage projects are controlled by the user roles and container ACLs. Based on the roles defined for the user, object users can be administrative users and non-administrative users. Non-admin users can do only operations per container based on the container's X-Container-Read and X-Container-Write ACLs. Container ACLs can be defined to limit access to objects in swift containers. Read access can be limited to allow download only, or allow download and listing. Write access allows the user to upload new objects to a container.

You can use an external Active Directory (AD) or Lightweight Directory Access Protocol (LDAP) server or a local database as the back-end to store and manage user credentials for user authentication. The authorization details such as relation of users with projects and roles are maintained locally by the keystone server. You can select the authentication server to be used. For example, if AD is existing in an enterprise deployment and the users in AD are required to access object data, you can decide to use AD as the back-end authentication server.

When the back-end authentication server is AD or LDAP, the user management operations such as creating a user and deleting a user are the responsibility of the AD or LDAP administrator, who can optionally also be the Keystone server administrator. When local authentication is used for object access, the user management operations are done by the Keystone administrator. With authorization, the management tasks such as creating roles, projects, and associating the user with them is done by the Keystone administrator. The Keystone administration can be done through the Keystone V3 REST API or by using an OpenStack python-based client.

Before you start creating object users, and projects, make sure that Keystone server is configured and the authentication servers are set up properly. Run the following command to see whether Keystone is configured properly:

```
mmces service list -a -v
```

The object users are authorized to the object data and resources by creating and managing roles and ACLs. The roles and ACLs define the actions that can be done by the user on the object resources such as accessing data, managing the projects, creating projects, read, write, and run permissions.

Related concepts

[Authorizing file protocol users](#)

The IBM Storage Scale system uses ACLs to authorize users who access the system through file protocols such as NFS and SMB.

[Authorization limitations](#)

Authorization limitations are specific to the protocols that are used to access data.

Configuring container ACLs to authorize object data users

The following examples and sections give an understanding on how to set up container ACLs and define the access permissions for the user.

Important:

- CES Swift Object protocol feature is not supported from IBM Storage Scale 5.2.0 onwards.
- IBM Storage Scale 5.1.8 is the last release that has CES Swift Object protocol.
- IBM Storage Scale 5.2.0 will tolerate the update of a CES node from IBM Storage Scale 5.1.8.
 - *Tolerate* means:
 - The CES node will be updated to 5.2.0.
 - Swift Object support will not be updated as part of the 5.2.0 update.
 - You may continue to use the version of Swift Object protocol that was provided in IBM Storage Scale 5.1.8 on the CES 5.2.0 node.
 - IBM will provide usage and known defect support for the version of Swift Object that was provided in IBM Storage Scale 5.1.8 until you migrate to a supported object solution that IBM Storage Scale provides.
 - CES Swift Object is replaced with IBM Storage Scale S3. For more details, refer the *S3 support overview* section the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
 - For more information about Swift Object in IBM Storage Scale, refer to the [IBM Storage Scale 5.2.0 documentation](#).
- Please contact IBM for further details and migration planning.

Creating containers

The Object Storage organizes data in account, container, and object. Each account and container is an individual database that is distributed across the cluster. An account database contains the list of containers in that account. A container database contains the list of objects in that container.

It is the responsibility of the Keystone server administrator to create and manage accounts. The account defines a namespace for containers. A container must be unique within the owning account and account must use a unique name within the project. The admin account is created by default.

To work with this function in the IBM Storage Scale GUI, log on to the GUI and select **Object > Containers**.

Use the following procedure to create containers:

1. Run the **swift post container** command to create a container by using the Swift command-line client.

In the following example, the Keystone administrator creates a `public_readOnly` container in `admin` account:

```
# swift post public_readOnly --os-auth-url http://tully-ces-ip.adcons.spectrum:35357/v3  
--os-project-name admin --os-project-domain-name Default --os-username admin  
--os-user-domain-name Default --os-password Passw0rd --auth-version 3
```

2. Run the following command to list the containers that are available for the account.

In the following example, the system lists the containers that are available in the `admin` project:

```
# swift list --os-auth-url http://tully-ces-ip.adcons.spectrum:35357/v3  
--os-project-name admin --os-project-domain-name Default --os-username admin  
--os-user-domain-name Default --os-password Passw0rd --auth-version 3  
public_readOnly
```

3. Run the following command to list the accounts, containers, or objects details.

In the following example, the system displays the `admin` account details:

```
# swift stat -v --os-auth-url http://tully-ces-ip.adcons.spectrum:35357/v3
--os-project-name admin --os-project-domain-name Default --os-username admin
--os-user-domain-name Default --os-password Passw0rd --auth-version 3
    StorageURL: http://tully-ces-ip.adcons.spectrum:8080/v1
/AUTH_bea5a0c632e54eaf85e9150a16c443ce
    Auth Token: 1f6260c4f8994581a465b8225075c932
    Account: AUTH_bea5a0c632e54eaf85e9150a16c443ce
    Containers: 1
    Objects: 0
    Bytes: 0
Containers in policy "policy-0": 1
Objects in policy "policy-0": 0
    Bytes in policy "policy-0": 0
    X-Account-Project-Domain-Id: default
        X-Timestamp: 1432766053.43581
        X-Trans-Id: tx9b96c4a8622c40b3ac69a-0055677ce7
        Content-Type: text/plain; charset=utf-8
        Accept-Ranges: bytes
```

In the following example, the system displays the public_readOnly' container details, on the admin account:

```
# swift stat public_readOnly -v --os-auth-url http://tully-ces-ip.adcons.spectrum:35357/v3
--os-project-name admin --os-project-domain-name Default --os-username admin
--os-user-domain-name Default --os-password Passw0rd --auth-version 3
URL: http://tully-ces-ip.adcons.spectrum:8080/v1/AUTH_bea5a0c632e54eaf85e9150a16c443ce
/public_readOnly
    Auth Token: 957d6c37155b44d3a476441bc927835d
    Account: AUTH_bea5a0c632e54eaf85e9150a16c443ce
    Container: public_readOnly
    Objects: 0
    Bytes: 0
    Read ACL:
    Write ACL:
    Sync To:
    Sync Key:
    Accept-Ranges: bytes
X-Storage-Policy: Policy-0
    X-Timestamp: 1432795292.10297
    X-Trans-Id: tx9b05c2135a9c4034b910c-0055677dad
    Content-Type: text/plain; charset=utf-8
```

By default, only users who are having a Keystone role that is specified in the proxy-server.conf operator_roles option are allowed to create container on an account.

Run the following command to list operator_roles on the IBM Storage Scale system during installation:

```
mmobj config list --ccrfile proxy-server.conf --section filter:keystoneauth --property
operator_roles
```

Run the following command to list operator_roles in all other cases:

```
mmobj config list --ccrfile proxy-server.conf --section filter:keystone --property
operator_roles
```

Keystone administrator can also use the container to control access to the objects by using an access control list (ACL). In the following example, a member of the admin account tries to display the details of public_readOnly account. However, the process fails because it does not have an operator role or access control defined:

```
# swift stat public_readOnly -v --os-auth-url http://tully-ces-ip.adcons.spectrum:35357/v3
--os-project-name admin --os-project-domain-name Default --os-username member
--os-user-domain-name Default --os-password Passw0rd --auth-version 3
Container HEAD failed: http://tully-ces-ip.adcons.spectrum:8080/v1
/AUTH_bea5a0c632e54eaf85e9150a16c443ce/public_readOnly 403 Forbidden
```

Related tasks

[“Creating read ACLs to authorize object users” on page 493](#)

The Keystone administrator can create container ACLs to grant read permissions using X-Container-Read headers in curl tool or --read-acl flag in the Swift command-line client.

[“Creating write ACLs to authorize object users” on page 495](#)

The Keystone administrator can create container ACLs to grant write permissions using X-Container-Write headers in the curl tool or --write-acl flag in the Swift command-line client.

Creating read ACLs to authorize object users

The Keystone administrator can create container ACLs to grant read permissions using X-Container-Read headers in curl tool or --read-acl flag in the Swift command-line client.

The following example shows how to create read permission in an ACL:

1. Upload the object *imageA.JPG* to *public_readOnly* container as the Keystone administrator.

```
# swift upload public_readOnly imageA.JPG --os-auth-url http://tully-ces-ip.adcons.spectrum:35357/v3  
--os-project-name admin --os-project-domain-name Default --os-username admin  
--os-user-domain-name Default --os-password Passw0rd --auth-version 3  
imageA.JPG
```

2. Issue the **swift post** command to provide public read access to the *public_readOnly* container.

```
# swift post public_readOnly --read-acl ".r:*,.rlistings" --os-auth-url http://tully-ces-ip.adcons.spectrum:35357/v3 --os-project-name admin --os-project-domain-name Default --os-username admin --os-user-domain-name Default --os-password Passw0rd --auth-version 3
```

Note: The *.r:** ACL specifies access for any referrer regardless of account affiliation or user name. The *.rlistings* ACL allows to list the containers and read (download) objects.

3. Issue the **swift stat** command at the container level to see the access details.

```
# swift stat public_readOnly -v --os-auth-url http://tully-ces-ip.adcons.spectrum:35357/v3  
--os-project-name admin --os-project-domain-name Default --os-username admin  
--os-user-domain-name Default --os-password Passw0rd --auth-version 3  
          URL: http://tully-ces-ip.adcons.spectrum:8080/v1/  
AUTH_bea5a0c632e54eaf85e9150a16c443ce  
/public_readOnly  
    Auth Token: 91a27a5ed8dc40d582e71844ca019c32  
      Account: AUTH_bea5a0c632e54eaf85e9150a16c443ce  
    Container: public_readOnly  
      Objects: 3  
        Bytes: 8167789  
    Read ACL: .r:*,.rlistings  
   Write ACL:  
     Sync To:  
     Sync Key:  
Accept-Ranges: bytes  
X-Trans-Id: tx73b0696705b94bf885bd5-0055678ab1  
X-Storage-Policy: Policy-0  
X-Timestamp: 1432795292.10297  
Content-Type: text/plain; charset=utf-8
```

4. As the *student* user from the *students* account, list and download the details of *public_readOnly* container that is created in the *admin* account.

Listing the details:

```
# swift stat public_readOnly -v --os-auth-url http://tully-ces-ip.adcons.spectrum:35357/v3  
--os-project-name students --os-project-domain-name Default --os-username student1  
--os-user-domain-name Default --os-password Passw0rd --auth-version 3 --os-storage-url  
http://tully-ces-ip.adcons.spectrum:8080/v1/AUTH_bea5a0c632e54eaf85e9150a16c443ce  
          URL: http://tully-ces-ip.adcons.spectrum:8080/v1/  
AUTH_bea5a0c632e54eaf85e9150a16c443ce  
/public_readOnly  
    Auth Token: d6ee0fb5e33748b1b9035a3b690c7587  
      Account: AUTH_bea5a0c632e54eaf85e9150a16c443ce  
    Container: public_readOnly  
      Objects: 3  
        Bytes: 8167789  
    Read ACL:  
   Write ACL:  
     Sync To:  
     Sync Key:
```

```

Accept-Ranges: bytes
X-Storage-Policy: Policy-0
X-Timestamp: 1432795292.10297
X-Trans-Id: tx09893920a6154faab6ace-0055678f6d
Content-Type: text/plain; charset=utf-8

```

Listing the container objects:

```

# swift list public_readOnly --os-auth-url http://tully-ces-ip.adcons.spectrum:35357/v3
--os-project-name students --os-project-domain-name Default --os-username student1
--os-user-domain-name Default --os-password Passw0rd --auth-version 3 --os-storage-url
http://tully-ces-ip.adcons.spectrum:8080/v1/AUTH_bea5a0c632e54eaf85e9150a16c443ce
file.txt
imageA.JPG
imageB.JPG

```

Downloading container objects:

```

# swift download public_readOnly --os-auth-url http://tully-ces-ip.adcons.spectrum:35357/v3
--os-project-name students --os-project-domain-name Default --os-username student1
--os-user-domain-name Default --os-password Passw0rd --auth-version 3 --os-storage-url
http://tully-ces-ip.adcons.spectrum:8080/v1/AUTH_bea5a0c632e54eaf85e9150a16c443ce
imageB.JPG [auth 0.321s, headers 0.380s, total 0.390s, 37.742 MB/s]
file.txt [auth 0.533s, headers 0.594s, total 0.594s, 0.000 MB/s]
imageA.JPG [auth 0.119s, headers 0.179s, total 18.135s, 0.308 MB/s]

```

- As the *student1* user from the *students* account, receive deny write access, while trying to upload new content in the *public_readOnly* container:

```

# swift upload public_readOnly photo.jpg --os-auth-url http://tully-ces-
ip.adcons.spectrum:35357/v3
--os-project-name students --os-project-domain-name Default --os-username student1
--os-user-domain-name Default --os-password Passw0rd --auth-version 3 --os-storage-url
http://tully-ces-ip.adcons.spectrum:8080/v1/AUTH_bea5a0c632e54eaf85e9150a16c443ce
Warning: failed to create container 'public_readOnly': 403 Forbidden:

Forbidden

Access was denied to this resource
Object PUT failed: http://tully-ces-ip.adcons.spectrum:8080/v1/
AUTH_bea5a0c632e54eaf85e9150a16c443ce
/public_readOnly/photo.jpg 403 Forbidden

```

Manipulating the read ACLs

The following table list different read ACLs combinations:

Table 43. ACL options that are available to manipulate object read ACLs	
Permission	Read ACL options
Read for all referrers	.r:*
Read and list for all referrers and listing	.r:*,rlistings
Read and list for a user in a specific project	<project_id>:<user_id>
Read and list for a user in every project	*:<user_id>
Read and list for every user in a project	<project_id>:<*>
Read and list for every user in every project	<*>:<*>

Note: In ACL settings, you must specify project IDs and user IDs rather than project names and user names. In a sequence of ACLs, separate the ACLs with commas: -read-acl c592e4f4:bdd3218,87c14a43:db2e994a.

Related tasks

[“Creating containers” on page 491](#)

The Object Storage organizes data in account, container, and object. Each account and container is an individual database that is distributed across the cluster. An account database contains the list of containers in that account. A container database contains the list of objects in that container.

[“Creating write ACLs to authorize object users” on page 495](#)

The Keystone administrator can create container ACLs to grant write permissions using X-Container-Write headers in the curl tool or `--write-acl` flag in the Swift command-line client.

Creating write ACLs to authorize object users

The Keystone administrator can create container ACLs to grant write permissions using X-Container-Write headers in the curl tool or `--write-acl` flag in the Swift command-line client.

Provides an example on how to configure write ACLs by using curl tool.

1. Run the following command to create a token:

```
token=$(openstack --os-auth-url http://tully-ces-ip.adcons.spectrum:35357/v3  
--os-project-name admin --os-project-domain-name Default --os-username admin  
--os-user-domain-name Default --os-password Passw0rd --os-identity-api-version 3  
token issue | grep -w "id" | awk '{print $4}')
```

2. Create a container that is named *writeOnly* with write permissions for a *member* user (with an ID of 4720614) who is part of the *admin* project (46b37eb) and a *student1* user (f58b7c09) who is part of the *students* project (d5c05730). In the X-Container-Write statement, you must specify the project and user IDs rather than the names:

```
# curl -i http://tully-ces-ip.adcons.spectrum:8080/v1/AUTH_bea5a0c632e54eaf85e9150a16c443ce  
/writeOnly -X PUT -H "Content-Length: 0" -H "X-Auth-Token: ${token}" -H  
"X-Container-Write: 46b37eb:4720614,d5c05730:f58b7c09" -H "X-Container-Read: "  
HTTP/1.1 201 Created  
Content-Length: 0  
Content-Type: text/html; charset=UTF-8  
X-Trans-Id: txf7b0bfef877345949c61c-005567b9d1  
Date: Fri, 29 May 2015 00:58:57 GMT
```

3. Issue a token as *student1* from the *students* project and upload an object by using the curl tool:

```
token=$(openstack --os-auth-url http://tully-ces-ip.adcons.spectrum:35357/v3  
--os-project-name students --os-project-domain-name Default --os-username student1  
--os-user-domain-name Default --os-password Passw0rd --os-identity-api-version 3  
token issue | grep -w "id" | awk '{print $4}')  
  
# curl -i http://tully-ces-ip.adcons.spectrum:8080/v1/AUTH_bea5a0c632e54eaf85e9150a16c443ce  
/writeOnly/imageA.JPG -X PUT -H "X-Auth-Token: ${token}" --upload-file imageA.JPG  
HTTP/1.1 100 Continue  
HTTP/1.1 201 Created  
Last-Modified: Fri, 29 May 2015 01:11:28 GMT  
Content-Length: 0  
Etag: 95d8c44b757f5b0c111750694dffef2b  
Content-Type: text/html; charset=UTF-8  
X-Trans-Id: tx6caa0570bfcd419782274-005567bcbe  
Date: Fri, 29 May 2015 01:11:28 GMT
```

4. List the state of the *writeOnly* container as *student1* user of the *students* project:

```
# curl -i http://tully-ces-ip.adcons.spectrum:8080/v1/AUTH_bea5a0c632e54eaf85e9150a16c443ce  
/writeOnly/imageA.JPG -X HEAD -H "X-Auth-Token: ${token}"  
HTTP/1.1 403 Forbidden  
Content-Type: text/html; charset=UTF-8  
X-Trans-Id: tx4f7dfbf7d4204785b6b50-005567bd8c  
Content-Length: 0  
Date: Fri, 29 May 2015 01:14:52 GMT
```

Note: This operation fails as the user does not have the necessary privileges.

5. Grant read permissions to *student1* user of the *students* project. In the X-Container-Read statement, you must specify the project and user IDs rather than the names:

```
token=$(openstack --os-auth-url http://tully-ces-ip.adcons.spectrum:35357/v3  
--os-project-name admin --os-project-domain-name Default --os-username admin  
--os-user-domain-name Default --os-password Passw0rd --os-identity-api-version 3
```

```

token issue | grep -w "id" | awk '{print $4}')

# curl -i http://tully-ces-ip.adcons.spectrum:8080/v1/AUTH_be
bea5a0c632e54eaf85e9150a16c443ce
/writeOnly -X POST -H "Content-Length: 0" -H "X-Auth-Token:
${token}" -H "X-Container-Read: d5c05730:f58b7c09"
HTTP/1.1 204 No Content
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Traffic-Id: tx77aafe0184da4b68a7756-005567beac
Date: Fri, 29 May 2015 01:19:40 GMT

```

- Verify whether the *sutdent1* user has the read access now:

```

token=$(openstack --os-auth-url http://tully-ces-ip.adcons.spectrum:35357/v3
--os-project-name students --os-project-domain-name Default --os-username student1
--os-user-domain-name Default --os-password Passw0rd --os-identity-api-version 3
token issue | grep -w "id" | awk '{print $4}')

# curl -i http://tully-ces-ip.adcons.spectrum:8080/v1/AUTH_be
bea5a0c632e54eaf85e9150a16c443ce
/writeOnly -X GET -H "X-Auth-Token: ${token}"
HTTP/1.1 200 OK
Content-Length: 11
X-Container-Object-Count: 1
Accept-Ranges: bytes
X-Storage-Policy: Policy-0
X-Container-Bytes-Used: 5552466
X-Timestamp: 1432861137.91693
Content-Type: text/plain; charset=utf-8
X-Traffic-Id: tx246b39018a5c4bcb90c7f-005567bff3
Date: Fri, 29 May 2015 01:25:07 GMT

imageA.JPG

```

Note: Object Storage does not support public write ACLs.

Related tasks

[“Creating containers” on page 491](#)

The Object Storage organizes data in account, container, and object. Each account and container is an individual database that is distributed across the cluster. An account database contains the list of containers in that account. A container database contains the list of objects in that container.

[“Creating read ACLs to authorize object users” on page 493](#)

The Keystone administrator can create container ACLs to grant read permissions using X-Container-Read headers in curl tool or –read-acl flag in the Swift command-line client.

Authorization limitations

Authorization limitations are specific to the protocols that are used to access data.

NFS ACL limitations

ACLs are stored as NFSv4 ACLs in the file system.

For more information about limitations of the NFSV4 ACLs, see [“Exceptions and limitations to NFS V4 ACLs support” on page 473](#).

SMB ACL limitations

The following are the SMB ACL limitations:

- ACL of a new child file or directory depends on the ACL type, the file system settings, and the ACL of the parent directory. Depending on these variables, the results in the IBM Storage Scale might be slightly different than in Microsoft Windows. For example, if the parent directory is set to have two ACEs, for example full access for owner and for everyone, the Windows default is to create two ACLs for the child. One is to allow full access for owner and other to allow full access for everyone. The IBM Storage Scale system by default creates six ACLs to allow and deny ACLs for owner, group, and everyone.
- If domain server manages the UID and GID mapping, the UID and GID mappings must be configured in the domain server before an ACE for that user or group can be created.

- Users and groups that belonged to another domain, and was migrated to a new domain by using the SID-History mechanism, cannot be stored in an ACL.
- Most well-known SIDs and built-in SIDs cannot be stored in an ACL. Only the "Everyone" SID can be stored and used in an IBM Storage Scale system.
- The SMB ACLs cannot be modified when LDAP-based authentication is used for file access.
- By using Microsoft Windows, you can limit the scope of inheritance for an ACE to one inheritance by selecting the **Apply these permissions to objects and/or containers within this container only** checkbox in the Windows Explorer. The IBM Storage Scale system does not support to configure this option and limit the scope of inheritance for an ACL.
- ACL inheritance stops at fileset junction points. New filesets always have the default ACL (770 root root).
- The root path of every SMB share needs read permission (read data, read attribute, read extended attribute) for everyone to prevent the unexpected behavior of, for example, Windows Explorer.
- To prevent display of Access Denied errors, the user must have the read attribute permission on all parent directories, when they have access to a file or directory.
- The value of the dacl_protected bit related to the Include Inheritable permissions from this object's parent checkbox can be changed only through SMB. The ACL commands cannot access this field. Setting a new ACL resets this field.
- The commands that are used to work on the ACLs do not support recursive updates of inherited ACEs in the file tree.
- Access privileges that are defined in Windows are not honored. Those privileges are tied to administrator groups and allow access, where the ACL alone does not grant it.
- Audit and alarm ACEs are not supported inside an ACL.
- The Bypass Traverse Check is implemented in GPFS for SMB clients only. Clients that use other protocols might still be locked out because the parent tree of an export has more restrictive ACLs than the export itself.
- POSIX-style ACLs are not supported.
- Similar to the POSIX standard, which is needed to read the content of a subdirectory, apart from the read permission in the ACL of this subdirectory, you also need to have traversal permission (SEARCH in Windows, EXECUTE in POSIX) for all of the parent directories. You can set the traverse permission in the "Everyone" group ACE at the share root, and inherit this privilege to all subdirectories. For the SMB protocol, this permission is applicable only if the *bypassTraversalCheck* configuration option is disabled.
- Even though the underlying file system does not enforce the permissions for extended attributes (READ_NAMED and WRITE_NAMED), these permissions are enforced for SMB clients.

ACL limitations that are applicable to all protocols

The following limitations are applicable to all protocols:

- When you create a file system, you need to specify -k nfs4 to specifically use NFSv4 ACLs, otherwise the default -k all uses both POSIX ACLs and NFSv4 ACLs.
- The IBM Storage Scale Object Storage does not do file share with NFS and SMB.

Related concepts

Authorizing file protocol users

The IBM Storage Scale system uses ACLs to authorize users who access the system through file protocols such as NFS and SMB.

Authorizing object users

The Object Storage service of the IBM Storage Scale system uses Keystone service for identity management. The identity management service consists of user authentication and authorization processes.

Chapter 37. Native NFS and GPFS

GPFS file systems may be exported using the Network File System (NFS) protocol from one or more nodes. After export, normal access to the file system can proceed from GPFS cluster nodes or NFS client nodes.

Note: GPFS on Windows does not provide NFS integration.

Considerations for the interoperability of a GPFS file system include:

- [“Exporting a GPFS file system using NFS” on page 499](#)
- [“NFS usage of GPFS cache” on page 502](#)
- [“Synchronous writing using NFS” on page 502](#)
- [“Unmounting a file system after NFS export” on page 502](#)
- [“NFS automount considerations” on page 503](#)
- [“Clustered NFS and GPFS on Linux” on page 503](#)

Note: None of these sections consider the NFS server integration that is introduced with CES. The integrated NFS server interactions, with the following documented sections, will be addressed in a future release.

Exporting a GPFS file system using NFS

To export a GPFS file system:

1. Create and mount the GPFS file system. In the examples, we assume a file system with a local mount point of /gpfs.

For performance reasons, some NFS implementations cache file information on the client. Some of the information (for example, file state information such as file size and timestamp) is not kept up-to-date in this cache. The client may view stale inode data (on `ls -l`, for example) if exporting a GPFS file system with NFS.

If this is not acceptable for a given installation, caching can be turned off by mounting the file system on the client using the appropriate operating system mount option (for example, `-o noac` on Linux NFS clients). Turning off NFS caching results in extra file systems operations to GPFS, and negatively affect its performance.

Note:

- Ensure that all GPFS file systems that you use to export data via NFS are mounted with the `syncnfs` option to prevent clients from running into data integrity issues during failover. Prior to mounting a GPFS system, it is a good practice to run the `mmchfs` command to set the `syncnfs` option, `-o syncnfs`.
- Ensure that NFS clients mount with the `-o hard` option to prevent any application failures during network failures or node failovers.
- If caching is turned on for the NFS clients, files that are migrated to the cloud storage tier by using Transparent cloud tiering remain in the co-resident status, and the capacity is not freed from the file system. However, if caching is disabled, the files are moved to the non-resident status and the capacity is freed. In this case, there is a negative impact on the performance. Therefore, there is a tradeoff between capacity and performance, and administrators must take a judicious decision depending on the business requirements.

2. Make sure that the clocks of all nodes in the GPFS cluster are synchronized. If this is not done, NFS access to the data, as well as other GPFS file system operations, may be disrupted.

NFS relies on metadata timestamps to validate the local operating system cache. If the same directory is either NFS-exported from more than one node, or is accessed with both the NFS and GPFS mount

point, it is critical that clocks on all nodes that access the file system (GPFS nodes and NFS clients) are constantly synchronized using appropriate software (for example, NTP). Failure to do so may result in stale information seen on the NFS clients.

3. Ensure that NFS is properly configured and running.

For Linux nodes, information on configuring NFS can be obtained at the [LinuxDocs.org](#).

For AIX nodes, refer to [AIX in IBM Documentation](#) for information about configuring NFS.

Related concepts

[NFS usage of GPFS cache](#)

[Synchronous writing using NFS](#)

[NFS automount considerations](#)

[Clustered NFS and GPFS on Linux](#)

Related tasks

[Unmounting a file system after NFS export](#)

Export considerations

Keep these points in mind when exporting a GPFS file system to NFS. The operating system being used and the version of NFS might require special handling or consideration.

Linux export considerations

Linux does not allow a file system to be NFS V4 exported unless it supports POSIX ACLs. For more information, see [“Linux ACLs and extended attributes” on page 474](#).

For Linux nodes only, issue the `exportfs -ra` command to initiate a reread of the `/etc/exports` file.

Starting with Linux kernel version 2.6, an `fsid` value must be specified for each GPFS file system that is exported on NFS. For example, the format of the entry in `/etc/exports` for the GPFS directory `/gpfs/dir1` might look like this:

```
/gpfs/dir1 cluster1(rw,fsid=745)
```

The administrator must assign `fsid` values subject to the following conditions:

1. The values must be unique for each file system.
2. The values must not change after reboots. The file system should be unexported before any change is made to an already assigned `fsid`.
3. Entries in the `/etc/exports` file are not necessarily file system roots. You can export multiple directories within a file system. In the case of different directories of the same file system, the `fsids` must be different. For example, in the GPFS file system `/gpfs`, if two directories are exported (`dir1` and `dir2`), the entries might look like this:

```
/gpfs/dir1 cluster1(rw,fsid=745)
/gpfs/dir2 cluster1(rw,fsid=746)
```

4. If a GPFS file system is exported from multiple nodes, the `fsids` should be the same on all nodes.

Configuring the directories for export with NFSv4 differs slightly from the previous NFS versions. To configure the directories, do the following:

1. Define the root of the overall exported file system (also referred to as the pseudo root file system) and the pseudo file system tree. For example, to define `/export` as the pseudo root and export `/gpfs/dir1` and `/gpfs/dir2` which are not below `/export`, run:

```
mkdir -m 777 /export /export/dir1 /export/dir2
mount --bind /gpfs/dir1 /export/dir1
mount --bind /gpfs/dir2 /export/dir2
```

In this example, **/gpfs/dir1** and **/gpfs/dir2** are bound to a new name under the pseudo root using the bind option of the mount command. These bind mount points should be explicitly unmounted after GPFS is stopped and bind-mounted again after GPFS is started. To unmount, use the **umount** command. In the preceding example, run:

```
umount /export/dir1; umount /export/dir2
```

2. Edit the **/etc(exports** file. There must be one line for the pseudo root with **fsid=0**. For the preceding example:

```
/export cluster1(rw,fsid=0)
/export/dir1 cluster1(rw,fsid=745)
/export/dir2 cluster1(rw,fsid=746)
```

The two exported directories (with their newly bound paths) are entered into the **/etc(exports** file.

Large installations with hundreds of compute nodes and a few login nodes or NFS-exporting nodes require tuning of the GPFS parameters **maxFilesToCache** and **maxStatCache** with the **mmchconfig** command.

This tuning is required for the GPFS token manager (file locking), which can handle approximately 1,000,000 files in memory. The token manager keeps track of a total number of tokens, which equals **5000 * number of nodes**. This will exceed the memory limit of the token manager on large configurations. By default, each node holds 5000 tokens.

For information about the default values of **maxFilesToCache** and **maxStatCache**, see the description of the **maxStatCache** attribute in the topic *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

In versions of IBM Storage Scale earlier than 5.0.2, the stat cache is not effective on the Linux platform unless the Local Read-Only Cache (LROC) is configured. For more information, see the description of the **maxStatCache** parameter in the topic *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

If you are running at SLES 9 SP 1, the kernel defines the **sysctl** variable **fs.nfs.use_underlying_lock_ops**, which determines whether the NFS lockd is to consult the file system when granting advisory byte-range locks. For distributed file systems like GPFS, this must be set to **true** (the default is **false**).

You can query the current setting by issuing the command:

```
sysctl fs.nfs.use_underlying_lock_ops
```

Alternatively, the **fs.nfs.use_underlying_lock_ops = 1** record can be added to **/etc/sysctl.conf**. This record must be applied after initially booting the node, and after each reboot, by issuing the command:

```
sysctl -p
```

Because the **fs.nfs.use_underlying_lock_ops** variable is currently not available in SLES 9 SP 2 or later, when NFS-exporting a GPFS file system, ensure that your NFS server nodes are at the SP 1 level (unless this variable is made available in later service packs).

For additional considerations when NFS exporting your GPFS file system, refer to *File system creation considerations* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

AIX export considerations

AIX does not allow a file system to be exported by NFS V4 unless it supports NFS V4 ACLs.

NFS usage of GPFS cache

Exporting a GPFS file system from a node may result in significant additional demands on the resources at that node. Depending on the number of NFS clients, their demands, and specific mount options, you may want to increase either one or both of the `maxFilesToCache` and `pagepool`.

See the `mmchconfig` command.

You may also choose to restrict the use of the NFS server node through the normal GPFS path and not use it as either a file system manager node or an NSD server.

Related concepts

[Synchronous writing using NFS](#)

[NFS automount considerations](#)

[Clustered NFS and GPFS on Linux](#)

Related tasks

[Exporting a GPFS file system using NFS](#)

[Unmounting a file system after NFS export](#)

Synchronous writing using NFS

With Linux, write operations are usually asynchronous. If synchronous writes are required over NFS, edit the `/etc/exports` file to include `sync,no_wdelay`.

Related concepts

[NFS usage of GPFS cache](#)

[NFS automount considerations](#)

[Clustered NFS and GPFS on Linux](#)

Related tasks

[Exporting a GPFS file system using NFS](#)

[Unmounting a file system after NFS export](#)

Unmounting a file system after NFS export

Because NFS use of a GPFS file system might result in a file being held, attempting to unmount a GPFS file system might return a `Device is busy` error. If this occurs, stop the NFS daemons before attempting to unmount the file system at the NFS server.

- For KNFS on Linux, issue this command:

```
/etc/rc.d/init.d/nfs stop
```

- On AIX, issue this command:

```
stopsrc -g nfs
```

NFS can be restarted after the unmount completes.

- For KNFS on Linux, issue this command:

```
/etc/rc.d/init.d/nfs start
```

- For AIX, issue this command:

```
startsrc -g nfs
```

Related concepts

[NFS usage of GPFS cache](#)

[Synchronous writing using NFS](#)

[NFS automount considerations](#)
[Clustered NFS and GPFS on Linux](#)

Related tasks

[Exporting a GPFS file system using NFS](#)

NFS automount considerations

The default file system type when using the automounter daemon is NFS. When the `-fstype` option is not specified, and the server is the local node, a soft-mount of the local directory is done at the desired mount point. JFS is assumed as the only handler of local directories. A GPFS file system local soft-mount does not work implicitly, since the mount request is passed to JFS which then produces an error. When specifying `-fstype mmfs` the local soft-mount works because the mount is then passed to GPFS instead of JFS.

A GPFS soft-mount does not automatically unmount. Setting `-fstype nfs3` causes the local server mounts to always go through NFS. This allows you to have the same **auto.map** file on all nodes whether the server is local or not, and the automatic unmount will occur. If you want local soft-mounts of GPFS file systems while other nodes perform NFS mounts, you should have different **auto.map** files on the different classes of nodes. This should improve performance on the GPFS nodes as they will not have to go through NFS.

Related concepts

[NFS usage of GPFS cache](#)
[Synchronous writing using NFS](#)
[Clustered NFS and GPFS on Linux](#)

Related tasks

[Exporting a GPFS file system using NFS](#)
[Unmounting a file system after NFS export](#)

Clustered NFS and GPFS on Linux

In addition to the traditional exporting of GPFS file systems using NFS, GPFS allows you to configure a subset of the nodes in the cluster to provide a highly available solution for exporting GPFS file systems via NFS.

The participating nodes are designated as Cluster NFS (CNFS) member nodes and the entire setup is frequently referred to as CNFS or CNFS cluster.

In this solution, all CNFS nodes export the same file systems to the NFS clients. When one of the CNFS nodes fails, the NFS serving load moves from the failing node to another node in the CNFS cluster. Failover is done using recovery groups to help choose the preferred node for takeover.

Related concepts

[NFS usage of GPFS cache](#)
[Synchronous writing using NFS](#)
[NFS automount considerations](#)

Related tasks

[Exporting a GPFS file system using NFS](#)
[Unmounting a file system after NFS export](#)

Chapter 38. Accessing a remote GPFS file system

Learn about accessing the files in a GPFS file system from another cluster.

The ability to access and mount GPFS file systems that are owned by other clusters in a network of sufficient bandwidth is accomplished by using the `mmauth`, `mmremotecluster`, and `mmremotefs` commands. Each site in the network is managed as a separate cluster, while allowing shared file system access.

The cluster that owns the file system is responsible for administering the file system and granting access to other clusters on a per cluster basis. After access to a particular file system is granted to nodes in another GPFS cluster, the nodes can mount the file system and perform data operations as if the file system was locally owned.

Each node in the GPFS cluster that requires access to another cluster's file system must be able to open a TCP/IP connection to every node in the other cluster.

Nodes in two separate remote clusters that mount the same file system are not required to be able to open a TCP/IP connection to each other. For example, if a node in `clusterA` mounts a file system from `clusterB`, and a node in `clusterC` wants to mount the same file system, nodes in `clusterA` and `clusterC` do not have to communicate with each other.

Each node in the GPFS cluster that require file system access must have one of the following:

- A virtual connection to the file system data through an NSD server (refer to [Figure 13 on page 505](#)).
- A physical connection to the disks containing file system data (refer to [Figure 14 on page 506](#)).

In this example, network connectivity is required from the nodes in `clusterB` to all the nodes in `clusterA` even if the nodes in `clusterB` can access the disks in `clusterA` directly.

Note: Even when remote nodes have direct connectivity to the SAN, they still use a connection through an NSD server for any NSDs that have been configured with Persistent Reserve (PR). If you want the remote nodes to access the disks through their direct connection to the SAN, you must ensure that PR is not enabled on the NSDs. See [“Enabling and disabling Persistent Reserve” on page 284](#).

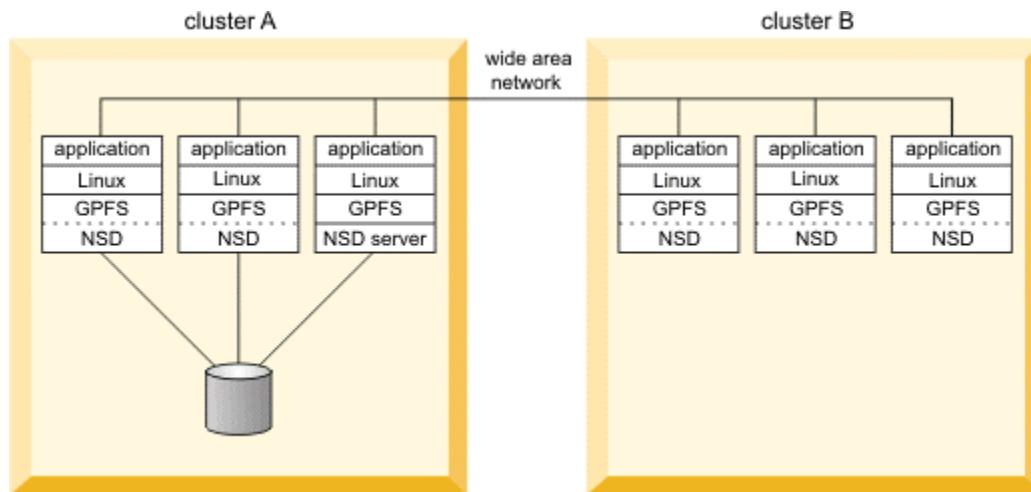


Figure 13. Remote mount of a file system by using NSD server access

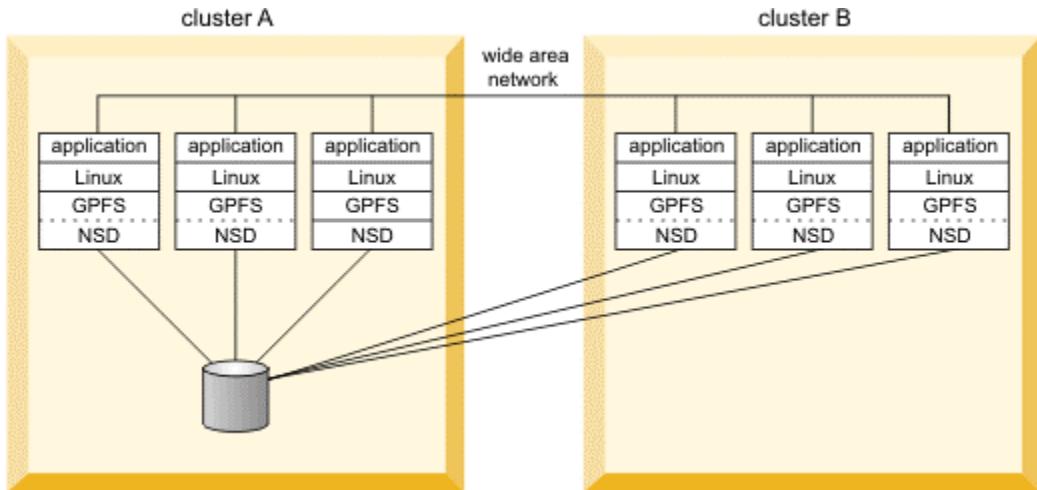


Figure 14. Remote mount of a file system by using SAN-attached disks

Note: SAN access of disks from different hardware platforms and operating systems are not supported. This limitation is because of the differences in how disks are labeled, how partitions are created and managed, and how multi-pathing managers react to error conditions among various operating systems and hardware platforms.

Note: If the cluster is configured with direct access to LUNs, then the disks might silently fall back to NSD server and results in the degradation of the network performance and slower I/O. For more information, see [“Changing NSD server usage and failback” on page 284](#).

Figure 15 on page 507 illustrates a multi-cluster configuration with multiple NSD servers. In this configuration:

- The two nodes in Cluster 1 are defined as the NSD servers (you can have up to eight NSD server nodes).
- All three clusters are connected with Gigabit Ethernet.
- Cluster 1 shares an InfiniBand switch network with Cluster 2 and an InfiniBand switch network with Cluster 3.

To take advantage of the fast networks and to use the nodes in Cluster 1 as NSD servers for Cluster 2 and Cluster 3, you must configure a subnet for each of the supported clusters. For example, issuing the command:

- `mmchconfig subnets=<IB_Network_1> <IB_Network_1>/Cluster1` in Cluster 2 allows nodes N₂ through N_x to use N₁ as an NSD server with InfiniBand Network 1 providing the path to the data.
- `mmchconfig subnets=<IB_Network_2> <IB_Network_2>/Cluster1` in Cluster 3 allows nodes N_{2+x} through N_{y+x} to use N_{1+x} as an NSD server with InfiniBand Network 2 providing the path to the data.

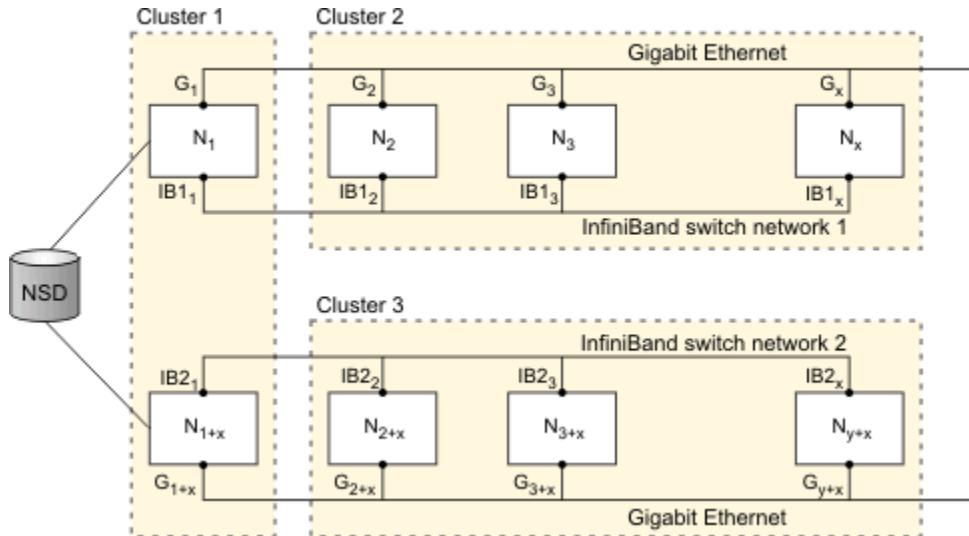


Figure 15. Multi-cluster configuration with multiple NSD servers

When you implement file access from other clusters, consider these topics:

- [“Remote user access to a GPFS file system” on page 507](#)
- [“Mounting a remote GPFS file system” on page 512](#)
- [“Managing remote access to a GPFS file system” on page 516](#)
- [“Using remote access with multiple network definitions” on page 518](#)
- [“Using multiple security levels for remote access” on page 520](#)
- [“Changing security keys with remote access” on page 521](#)
- [“Important information about remote access” on page 523](#)

Remote user access to a GPFS file system

In a cluster environment that has a single user identity namespace, all nodes have user accounts set up in a uniform manner. This is usually accomplished by having equivalent /etc/passwd and /etc/group files on all nodes in the cluster.

For consistency of ownership and access control, a uniform user identity namespace is preferred. For example, if user Jane Doe has an account on nodeA with the user name **janedoe** and user ID **1001** and group ID **500**, on all other nodes in the same cluster Jane Doe will have an account with the same user and group IDs. GPFS relies on this behavior to perform file ownership and access control tasks.

If a GPFS file system is being accessed from a node belonging to another GPFS cluster, the assumption about the uniform user account infrastructure might no longer be valid. Since different clusters can be administered by different organizations, it is possible for each of the clusters to have a unique set of user accounts. This presents the problem of how to permit users to access files in a file system owned and served by another GPFS cluster. In order to have such access, the user must be somehow known to the other cluster. This is usually accomplished by creating a user account in the other cluster, and giving this account the same set of user and group IDs that the account has in the cluster where the file system was created.

To continue with this example, Jane Doe would need an account with user ID **1001** and group ID **500** created in every other GPFS cluster from which remote GPFS file system access is desired. This approach is commonly used for access control in other network file systems, (for example, NFS or AFS®), but might pose problems in some situations.

For example, a problem arises if Jane Doe already has an account in some other cluster, but the user ID associated with this account is not **1001**, and another user in the other cluster has user ID **1001**. It would require a considerable effort on the part of system administrator to ensure that Jane Doe's account has the same set of IDs on all clusters. It is more desirable to be able to use the existing accounts without

having to make changes. GPFS helps to solve this problem by optionally performing user ID and group ID remapping internally, using user-supplied helper applications. For a detailed description of the GPFS user ID remapping convention, see the IBM white paper *UID Mapping for GPFS in a Multi-cluster Environment* (https://www.ibm.com/docs/en/storage-scale?topic=STXKQY/uid_gpfs.pdf).

Access from a remote cluster by a root user presents a special case. It is often desirable to disallow root access from a remote cluster while allowing regular user access. Such a restriction is commonly known as root squash. A root squash option is available when making a file system available for mounting by other clusters using the mmauth command. This option is similar to the NFS root squash option. When enabled, it causes GPFS to squash superuser authority on accesses to the affected file system on nodes in remote clusters.

This is accomplished by remapping the credentials: user id (UID) and group id (GID) of the root user, to a UID and GID specified by the system administrator on the home cluster, for example, the UID and GID of the user nobody. In effect, root squashing makes the root user on remote nodes access the file system as a non-privileged user.

Although enabling root squash is similar to setting up UID remapping, there are two important differences:

1. While enabling UID remapping on remote nodes is an option available to the remote system administrator, root squashing need only be enabled on the local cluster, and it will be enforced on remote nodes. Regular UID remapping is a user convenience feature, while root squashing is a security feature.
2. While UID remapping requires having an external infrastructure for mapping between local names and globally unique names, no such infrastructure is necessary for enabling root squashing.

When both UID remapping and root squashing are enabled, root squashing overrides the normal UID remapping mechanism for the root user.

Using NFS/SMB protocol over remote cluster mounts

IBM Storage Scale allows you to create NFS and SMB exports on remotely mounted file systems.

The following diagram shows the high-level flow of this feature.

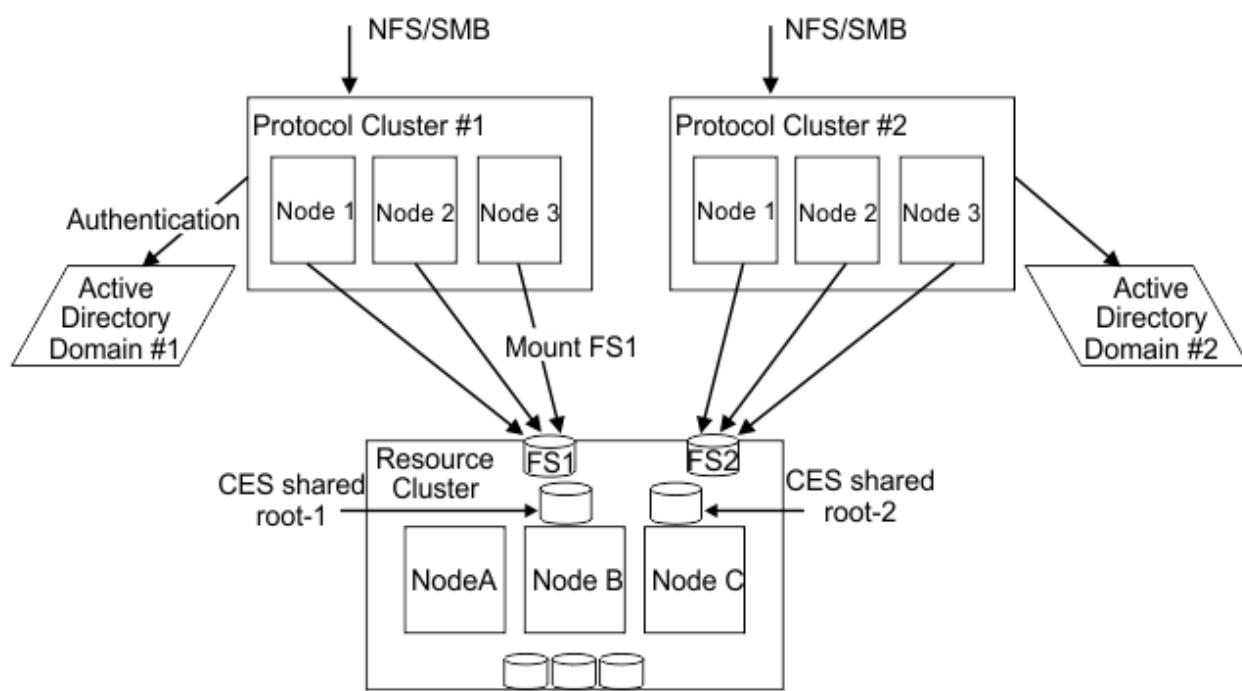


Figure 16. High-level flow of protocols on remotely mounted file systems

b11adv009

This allows you to separate the tasks performed by each cluster. Storage cluster owns the file systems and the storage. Protocol clusters contain the protocol node that provides access to the remotely mounted file system through NFS or SMB. In this configuration, each cluster is managed independently. For more information, see [“Important information about remote access” on page 523](#).

Here, the storage cluster owns a file system and the protocol cluster remotely mounts the file system. The protocol nodes (CES nodes) in the protocol cluster export the file system via SMB and NFS.

You can define one set of protocol nodes per cluster, using multiple independent protocol clusters which remotely mount file systems. Protocol clusters can share access to a storage cluster but not to a file system. Each protocol cluster requires a dedicated file system. Each protocol cluster can have a different authentication configuration, thus allowing different authentication domains while keeping the data at a central location. Another benefit is the ability to access existing ESS-based file systems through NFS or SMB without adding nodes to the ESS cluster.

Configuring protocols on a separate cluster

The process for configuring Cluster Export Services (CES) in a multi-cluster environment in many respects is the same as for a single cluster, however, there are a few differences mainly in the procedure order.

This procedure assumes an environment with the server, network, storage, and operating systems are installed and ready for IBM Storage Scale. For more information, see *Installing IBM Storage Scale on Linux nodes and deploying protocols in IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Do the following steps:

1. Install IBM Storage Scale on all nodes that are in the storage and protocol clusters.

Note: If you install a protocol cluster into an environment with an existing IBM Storage Scale cluster, the IBM Storage Scale version used should comply with the protocol cluster. The installation can be performed either manually or by using the installation toolkit. Do not create clusters or file systems or Cluster Export Services yet.

2. Create the storage and protocol clusters.

Note: Proceed with cluster creation of the storage cluster and one or more protocol clusters. Ensure that the configuration parameter maxBlockSize is set to the same value on all clusters.

3. Create file systems on the storage cluster, taking the following into consideration:

- CES shared root file system – Each protocol cluster requires its own CES shared root file system. Having a shared root file system that is different from the file system that serves data eases the management of CES.
- Data file systems – At least one file system is required for each protocol cluster configured for Cluster Export Services. A data file system can only be exported from a single protocol cluster.

4. Before installing and configuring Cluster Export Services, consider the following points:

- Authentication - Separate authentication schemes are supported for each CES cluster.
- ID mapping - The ID mapping of users authenticating to each CES cluster. It is recommended to have unique ID mapping across clusters, but not mandatory.

Note: You must judiciously determine the ID mapping requirements and prevent possible interference or security issues.

- GUI - GUI support for remote clusters is limited. Each cluster should have its own GUI. The GUI may be installed onto CES nodes but performance must be taken into consideration.
- Object - Object is not supported in multi-cluster configurations.
- For a list of limitations, see [“Limitations of protocols on remotely mounted file systems” on page 511](#).

5. Configure clusters for remote mount. For more information, see .

6. Install and configure Cluster Export Services by using the installation toolkit or manually. For more information, see *Installing IBM Storage Scale on Linux nodes with the installation toolkit* and *Manually*

installing the IBM Storage Scale software packages on Linux nodes in IBM Storage Scale: Concepts, Planning, and Installation Guide.

Note: Use the remotely mounted CES shared root file system.

7. Once SMB and/or NFS is enabled, new exports can be created on the remotely mounted data file system.

Managing multi-cluster protocol environments

In multi-cluster protocol environments, each cluster is managed independently, and there is no central management for all clusters that are included in IBM Storage Scale. You can centrally manage multiple IBM Storage Scale clusters by using Spectrum Control.

Consider the following aspects while you manage a multi-cluster protocol environment:

- Each cluster requires its own GUI. The GUI might be installed onto the CES nodes but performance must be taken into consideration.
- Each cluster has its own Rest API.
- Each cluster has its own health monitoring. This means that error events that are raised in the storage cluster are not visible in the protocol cluster and vice versa.
- Availability of certain performance metrics depends on the role of the cluster. That is, NFS metrics are available on protocol clusters only.
- Each cluster is installed and upgraded independently.

Due to the separation of duties (storage clusters own the file systems and protocol clusters own the NFS/SMB exports and S3 accounts and buckets) certain management tasks must be done in the corresponding cluster:

- File system-related operations like creating file systems, filesets, or snapshots must be done in the storage cluster.
- Export-related operations like creating exports, managing CES IP addresses, and managing authentication must be done in the protocol cluster.

Since the resource cluster is unaware of the authentication setup and UID mapping, all actions that require a user or a group name must be done in the corresponding protocol cluster (for example, generate quota reports and manage ACLs).

Upgrading multi-cluster environments

There is no special process to upgrade clusters in a multi-cluster environment. When you choose an IBM Storage Scale version, the release should comply with release level limitations.

Note: Upgrades are performed on a cluster-boundary basis.

Once all the clusters in the environment are upgraded, the release and the file system version should be changed. To view the differences between file system versions, see “[Listing file system attributes](#)” on page 228.

To change the IBM Storage Scale release, issue the following command on each cluster:

```
mmchconfig release=LATEST
```

Note: Nodes that run an older version of IBM Storage Scale on the remote cluster will no longer be able to mount the file system. Command fails if any nodes running an older version are mounted at time command is issued.

To change the file system version, issue the following command for each file system on the storage cluster:

```
mmchfs <fs> -V full
```

If your requirements call for it, issue the following command:

```
mmchfs <fs> -V compat
```

This enables only backward-compatible format changes.

Limitations of protocols on remotely mounted file systems

You must consider certain restrictions when you plan on setting up a multi-cluster protocol environment.

Refer to the following points:

- You can configure one storage cluster and up to five protocol clusters (current limit).
- The storage cluster owns all of the exported IBM Storage Scale file systems. This means at least two file systems per protocol cluster (one CES shared root + one data file system).
- The storage cluster must not have any protocol nodes (CES must be disabled).
- The protocol clusters cannot own any IBM Storage Scale file systems, only remote mounts from the storage cluster are allowed.
- Any file system can be remotely mounted by exactly one protocol cluster. Sharing a file system between multiple protocol clusters might cause data inconsistencies.
- The primary use case for multi-cluster protocol is to allow multiple authentication configurations. The setup must not be used for extending the scalability of Cluster Export Services (CES) or to work around defined limitations (for example, number of SMB connections).
- This setup provides some level of isolation between the clusters, but there is no strict isolation of administrative operations, and there is no guarantee that administrators on one cluster cannot see data from another cluster. Strict isolation is guaranteed through NFS or SMB access only.
- Each protocol cluster must use a dedicated file system, it is not allowed to share a file system between multiple protocol clusters.
- Storage and protocol clusters are in the same site/location, high network latencies between them can cause problems.
- This setup cannot be used for Object or iSCSI services.
- Due to the separation of duties (resource clusters own the file systems and protocol clusters own the NFS/SMB exports and S3 accounts and buckets), certain management task must be done in the corresponding cluster:
 - File system related operations (like creating file systems, filesets, creating snapshots) must be done in the resource cluster.
 - Export-related operations (creating exports, managing CES IPs, managing authentication and ACLs) must be done in the protocol cluster.

Note: This also means that certain operations such as creation of fileset and snapshots do not work on the GUI.

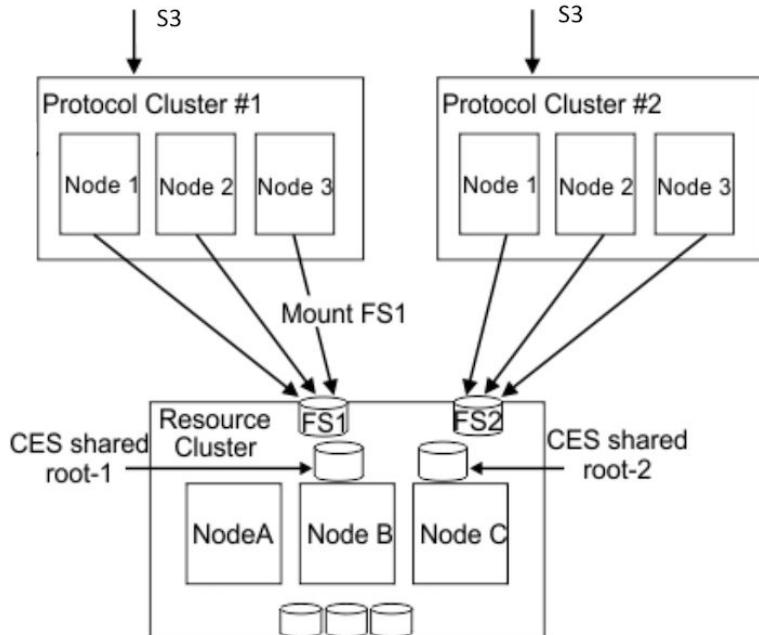
- Remote file systems are not mounted automatically resulting in client errors when a CES node on a protocol node is restarted.
- Cross-protocol change notifications will not work on remotely-mounted file systems. For example, if an NFS client changes a file, the system will not issue a "file change" notification to the SMB client which has asked for a notification.
- If you enable HDFS protocol, see the *Planning and Hadoop ACL and IBM Storage Scale protocols* sections in the *IBM Storage Scale: Big Data and Analytics Guide*.

S3 protocol over remote cluster mounts

IBM Storage Scale allows you to create S3 buckets on remotely mounted file systems.

The following figure shows the high-level flow of this feature:

Figure 17. High-level flow of protocols on remotely mounted file systems



This allows you to separate the tasks performed by each cluster. The IBM Storage Scale cluster owns the file systems and the storage. Protocol clusters contain the protocol node that provides access to the remotely mounted file system through S3. In this configuration, each cluster is managed independently. For more information, see [“Important information about remote access” on page 523](#).

Here, the resource cluster owns a file system and the protocol cluster remotely mounts the file system. The protocol nodes (CES nodes) in the protocol cluster make the file system available via S3 buckets.

You can define one set of protocol nodes per cluster, using multiple independent protocol clusters, which remotely mount file systems. Protocol clusters can share access to a resource cluster but not to a file system. Each protocol cluster requires a dedicated file system. Another benefit is the ability to access existing IBM Storage Scale System-based file systems through S3 without adding nodes to the IBM Storage Scale System cluster.

Mounting a remote GPFS file system

Explore an example of how to mount a file system that is owned and served by another IBM Storage Scale cluster.

The package `gpfs.gskit` must be installed on all the nodes of the owning cluster and the accessing cluster. For more information, see the installation chapter for your operating system, such as *Installing GPFS on Linux node and deploying protocols* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

The procedure to set up remote file system access involves the generation and exchange of authorization keys between the two clusters. In addition, the administrator of the GPFS cluster that owns the file system needs to authorize the remote clusters that are to access it, while the administrator of the GPFS cluster that seeks access to a remote file system needs to define to GPFS the remote cluster and file system whose access is desired.

Note: For more information about CES cluster setup, see [“CES cluster setup” on page 651](#).

In this example, `owningCluster` is the cluster that owns and serves the file system to be mounted and `accessingCluster` is the cluster that accesses `owningCluster`.

Note:

- The following example uses AUTHONLY as the authorization setting. When you specify AUTHONLY for authentication, GPFS checks network connection authorization. However, data sent over the connection is not protected.
- Clusters that are created on IBM Storage Scale version 4.2 or later are already created with AUTHONLY as the authentication mode. If the authentication mode used for `owningCluster` is AUTHONLY or a cipher other than empty, skip steps “1” on page 513 and “2” on page 513. If the authentication mode used for `accessingCluster` is AUTHONLY or a cipher other than empty, skip steps “4” on page 513 and “5” on page 513. You can use the **mmlsconfig cipherList** command to list the current cipher list that is being used by the local cluster.

1. On `owningCluster`, the system administrator issues the **mmauth genkey** command to generate a public/private key pair. The key pair is placed in `/var/mmfs/ssl`. The public key file is `id_rsa.pub`.

```
mmauth genkey new
```

2. On `owningCluster`, the system administrator enables authorization by entering the following command:

```
mmauth update . -1 AUTHONLY
```

3. The system administrator of `owningCluster` gives the file `/var/mmfs/ssl/id_rsa.pub` to the system administrator of `accessingCluster`. This operation requires the two administrators to coordinate their activities and must occur outside of the GPFS command environment.

The system administrator of `accessingCluster` can rename the key file and put it in any directory of the node that the administrator is working on, so long as the administrator provides the correct path and file name in the **mmremotecluster add** command in Step 9. In this example, the system administrator renames the key file to `owningCluster_id_rsa.pub`.

4. On `accessingCluster`, the system administrator issues the **mmauth genkey** command to generate a public/private key pair. The key pair is placed in `/var/mmfs/ssl`. The public key file is `id_rsa.pub`.

```
mmauth genkey new
```

5. On `accessingCluster`, the system administrator enables authorization by entering the following command:

```
mmauth update . -1 AUTHONLY
```

6. The system administrator of `accessingCluster` gives key file `/var/mmfs/ssl/id_rsa.pub` to the system administrator of `owningCluster`. This operation requires the two administrators to coordinate their activities, and must occur outside of the GPFS command environment.

The system administrator of `owningCluster` can rename the key file and put it in any directory of the node that they are working on, so long as the administrator provides the correct path and file name in the **mmauth add** command in Step 7. In this example, the system administrator renames the key file to `accessingCluster_id_rsa.pub`.

7. On `owningCluster`, the system administrator issues the **mmauth add** command to authorize `accessingCluster` to mount file systems that are owned by `owningCluster` by using the key file that was received from the administrator of `accessingCluster`:

```
mmauth add accessingCluster -k accessingCluster_id_rsa.pub
```

8. On `owningCluster`, the system administrator issues the **mmauth grant** command to authorize `accessingCluster` to mount specific file systems that are owned by `owningCluster`:

```
mmauth grant accessingCluster -f gpfs
```

where:

gpfs

Is the device name for the file system in `owningCluster`.

To see the device name, issue the `df -h` command, which displays the device name of the GPFS file system, as shown in the following example:

```
# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        1.9G   0    1.9G  0% /dev
tmpfs          1.9G   0    1.9G  0% /dev/shm
tmpfs          1.9G  17M  1.9G  1% /run
tmpfs          1.9G   0    1.9G  0% /sys/fs/cgroup
/dev/vda1       25G  1.5G  24G  6% /
GPFS           300G  2.5G 298G  1% /gpfs
tmpfs          379M   0   379M  0% /run/user/0
```

Note: If the accessing cluster is mounting the remote file system in read-only mode, only a subset of the events will be generated. For more information, see *File audit logging events' descriptions* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

9. On `accessingCluster`, the system administrator must define the cluster name, contact nodes and public key for `owningCluster`:

```
mmremotecluster add owningCluster -n node1,node2,node3 -k owningCluster_id_rsa.pub
```

This command provides the system administrator of `accessingCluster` a means to locate the serving cluster and mount its file systems.

10. On `accessingCluster`, the system administrator issues one or more `mmremotefs` commands to identify the file systems in `owningCluster` that are to be accessed by nodes in `accessingCluster`:

```
mmremotefs add mygpfs -f gpfs -C owningCluster -T /mygpfs
```

where:

mygpfs

Is the device name under which the file system is known in `accessingCluster`.

gpfs

Is the device name for the file system in `owningCluster`.

owningCluster

Is the name of `owningCluster` as given by the `mmlscluster` command on a node in `owningCluster`.

/mygpfs

Is the local mount point in `accessingCluster`.

11. On `accessingCluster`, the system administrator enters the `mmmount` command to mount the file system:

```
mmmount mygpfs
```

Table 44 on page 515 summarizes the commands that the administrators of the two clusters need to issue so that the nodes in `accessingCluster` can mount the remote file system `fs1`, which is owned by `owningCluster`, assigning `rfs1` as the local name with a mount point of `/rfs1`.

Table 44. Summary of commands to set up cross-cluster file system access.

accessingCluster	owningCluster
<code>mmauth genkey new</code>	<code>mmauth genkey new</code>
<code>mmauth update . -l AUTHONLY</code>	<code>mmauth update . -l AUTHONLY</code>
Note: The <code>mmauth genkey new</code> and <code>mmauth update</code> commands can be skipped if the given cluster (accessingCluster or owningCluster) is already operating with the authentication mode AUTHONLY (it is the default authentication mode on the clusters that are created on version 4.2 or later) or a cipher other than empty.	
Exchange public keys (file <code>/var/mmfs/ssl/id_rsa.pub</code>)	Exchange public keys (file <code>/var/mmfs/ssl/id_rsa.pub</code>)
<code>mmremotecluster add owningCluster ...</code>	<code>mmauth add accessingCluster ...</code>
<code>mmremotefs add rfs1 -f fs1 -C owningCluster -T /rfs1</code>	<code>mmauth grant accessingCluster -f fs1 ...</code>

Note: The configuration of remote clusters might impact the notify RPCs that are related to deadlock detection and amelioration, node overload, and node expel. The authentication of the notify RPCs is controlled by the value of the **sdrNotifyAuthEnabled** configuration parameter, which is local to a cluster. However, the notify RPCs can be used between remote clusters. Hence, it is recommended that remote clusters have a consistent setting of **sdrNotifyAuthEnabled** to avoid failures. For example, if the home cluster was created prior to version 5.1.1, it may have the **sdrNotifyAuthEnabled** configuration parameter set to no. If a new client cluster is created with version 5.1.1 or later, and that cluster has **sdrNotifyAuthEnabled** set to yes, then either the value should be set to yes on the home cluster (after the nodes of the home cluster are upgraded to version 5.1.1 or later), or the value should be set to no on the client cluster.

For more information about how to configure the **sdrNotifyAuthEnabled** parameter, see *mmchconfig command* in *IBM Storage Scale: Command and Programming Reference Guide*.

Fileset access control for remote clusters

In IBM Storage Scale, administrators can allow access to remote cluster nodes to only a subset of filesets instead of the entire file system.

The **mmauth grant** is issued to specify the list of allowed filesets while granting access to the file system. For example,

```
mmauth grant accessingCluster -f gpfs --fileset FilesetList
```

The **mmauth grant** command or the **mmauth deny** command can be issued to modify the list of allowed filesets. For example,

```
mmauth deny accessingCluster -f gpfs --fileset FilesetList
```

When access is allowed to only certain filesets, a subset of the file system namespace containing the allowed filesets is visible on the remote cluster nodes. The **mmlsfileset**, **mmlssnapshot** and **mmlsquota** command outputs will also display the corresponding information only on the allowed filesets.

Administrators must consider the following factors while setting up fileset access control for remote clusters:

- While setting up the allowed fileset list for the first time, the root fileset must be included.

- The root fileset cannot be removed from the allowed list later by using the **mmauth deny** command.
- Both independent and dependent filesets can be included in the allowed fileset list.
- When the allowed fileset list is created, the filesets that are not in that list cannot be accessed from the authorized remote clusters.
- If a fileset exists in the allowed list, all the filesets that are linked above it in the file system namespace must also exist in the list to ensure that the access to the fileset works properly.

Fileset access control does not offer a hard security boundary. It is therefore important that the root administrator of the remote cluster must be fully trusted by the root administrator of the home cluster.

For more information, see *mmauth command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Managing remote access to a GPFS file system

This is an example of how to manage remote access to GPFS file systems.

To see a list of all clusters authorized to mount file systems owned by `cluster1`, the administrator of `cluster1` issues this command:

```
mmauth show
```

To authorize a third cluster, say `cluster3`, to access file systems owned by `cluster1`, the administrator of `cluster1` issues this command:

```
mmauth add cluster3 -k cluster3_id_rsa.pub  
mmauth grant cluster3 -f /dev/gpfs1
```

To subsequently revoke `cluster3` authorization to access a specific file system `gpfs1` owned by `cluster1`, the administrator of `cluster1` issues this command:

```
mmauth deny cluster3 -f /dev/gpfs1
```

To completely revoke `cluster3` authorization to access file systems owned by `cluster1`, the administrator of `cluster1` issues this command:

```
mmauth delete cluster3
```

Note: Access controls such as grant and deny do not apply on the remote nodes that have the specified file system currently mounted. The new access control will be applied on the next mount of the file system.

Attaching direct storage on IBM Z

Direct storage attachment can lead to an I/O performance improvement on IBM Z Systems. This can be achieved by setting up a physical connection to the disks that contain file system data for each node in the participating clusters. The I/O workload is routed automatically by IBM Storage Scale to the directly attached storage using the SAN over a Fibre Channel (FC) network, which also reduces load on the cluster IP network.

IBM Storage Scale on IBM Z supports several types of shared backend storage like Direct Access Storage Devices (DASD) or SCSI devices. For more information about how to set up those devices for shared access, see [Getting started with IBM Storage Scale for Linux on IBM Z](#).

For ECKD-DASDs, by default, if all paths to the disk are unavailable, then the corresponding device in Linux waits for one of the paths to recover. I/O requests are blocked while the device is waiting. The `failfast` attribute of ECKD-DASDs can be set to immediately return the I/O requests as failed, while no path to the device is available. IBM Storage Scale then performs a failover of the I/O traffic to the network connection. For more information on how to set the `failfast` attribute, see [Enabling and disabling immediate failure of I/O requests and chzdev - Configure IBM Z devices](#).

When you use SCSI devices, a timeout can be set before the error recovery. The default timeout is 30 seconds. For more information about using SCSI devices, [Working with SCSI devices](#). Similar to the ECKD-DASDs, IBM Storage Scale also handles the path failover and fallback between SAN path and network connection.

You can run the **mmdiag --iohist** command to verify the usage of the direct storage path, here for the ECKD-DASD backend storage. If direct storage path is used for I/O, the column "Type" displays **lcl** for local, and the column "Device/NSD ID" displays the device name and the "NSD node" column must remain empty as shown in the following example:

```
# mmdiag --iohist
== mmdiag: iohist ==
I/O history:
I/O start time RW Buf type disk:sectorNum nSec time Type Device/NSDID NSD node
-----
09:02:45.109249 W data 3:19736896 512 1.110 lcl dasdd1
09:02:45.113789 W data 4:42704704 512 1.097 lcl dasde1
09:02:45.118548 W data 1:16916288 512 1.136 lcl dasdb1
09:02:45.123191 W data 5:21751616 512 1.219 lcl dasda1
09:02:45.127730 W data 2:16916288 512 1.262 lcl dasdc1
09:02:45.132335 W data 3:29004608 512 1.170 lcl dasdd1
09:02:45.136838 W data 4:3216192 512 1.044 lcl dasde1
09:02:45.141494 W data 1:17722176 512 1.132 lcl dasdb1
09:02:45.146166 W data 5:28198720 512 1.048 lcl dasda1
09:02:45.150742 W data 2:10872128 512 1.033 lcl dasdc1
09:02:45.155404 W data 3:17319232 512 1.120 lcl dasdd1
09:02:45.159955 W data 4:16110400 512 1.039 lcl dasde1
```

In case of an I/O failover from SAN path to the network connection, the output of the **mmdiag --iohist** command displays NSD IDs instead of the device names and it also displays the IP address of the NSD server node. The column "Type" now displays **cli** instead of **lcl** as shown in the following example:

```
# mmdiag --iohist
== mmdiag: iohist ==
I/O history:
I/O start time RW Buf type disk:sectorNum nSec time ms Type Device/NSD ID NSD
node
-----
17:58:34.489275 W logData 4:21198704 8 0.450 cli 9113AC14:60D9C779
192.168.22.220
17:58:34.719027 W logData 4:21198704 8 0.338 cli 9113AC14:60D9C779
192.168.22.220
17:58:34.720150 W logData 4:21198704 8 0.308 cli 9113AC14:60D9C779
192.168.22.220
17:58:34.721608 W logData 4:21198704 8 0.251 cli 9113AC14:60D9C779
192.168.22.220
17:58:36.039115 W logData 4:21198704 8 0.589 cli 9113AC14:60D9C779
192.168.22.220
17:58:36.042718 W iallocSeg 4:22204032 16 0.674 cli 9113AC14:60D9C779
192.168.22.220
17:58:36.042694 W iallocSeg 3:22204464 16 0.698 cli 9113AC14:60D9C778
192.168.22.220
17:58:36.043405 W logDesc 4:21198664 8 0.279 cli 9113AC14:60D9C779
192.168.22.220
18:00:58.540496 R metadata 3:21198656 512 1.218 cli 9113AC14:60D9C778
192.168.22.220
18:00:59.096484 R inode 2:21701464 8 0.478 cli 9113AC14:60D9C777
192.168.22.220
18:00:59.486800 W logData 4:21198704 8 0.493 cli 9113AC14:60D9C779
192.168.22.220
18:00:59.487349 W inode 2:21701464 8 0.421 cli 9113AC14:60D9C777
192.168.22.220
```

Using remote access with multiple network definitions

GPFS permits the use of both public and private IP address. Private IP addresses are typically used to communicate on private networks.

Private IP addresses are on one of these subnets:

- 10.0.0.0
- 172.16.0.0
- 192.168.0.0

See [RFC 1597 - Address Allocation for Private Internets](http://www.ip-doc.com/rfc/rfc1597) (www.ip-doc.com/rfc/rfc1597) for more information.

Use the `mmchconfig` command, `subnets` attribute, to specify the private IP addresses to be accessed by GPFS.

Figure 18 on page 520 describes an AIX cluster named CL1 with nodes named CL1N1, CL1N2, and so forth, a Linux cluster named CL2 with nodes named CL2N1, CL2N2, and another Linux cluster named CL3 with a node named CL3N1. Both Linux clusters have public Ethernet connectivity, and a Gigabit Ethernet configured with private IP addresses (10.200.0.1 through 10.200.0.24), not connected to the public Ethernet. The InfiniBand Switch on the AIX cluster CL1 is configured using public IP addresses on the 7.2.24/13 subnet and is accessible from the outside.

With the use of both public and private IP addresses for some of the nodes, the setup works as follows:

1. All clusters must be created using host names or IP addresses that correspond to the public network.
2. Using the `mmchconfig` command for the CL1 cluster, add the attribute: `subnets=7.2.24.0`.

This allows all CL1 nodes to communicate using the InfiniBand Switch. Remote mounts between CL2 and CL1 will use the public Ethernet for TCP/IP communication, since the CL2 nodes are not on the 7.2.24.0 subnet.

GPFS assumes subnet specifications for private networks are independent between clusters (private networks are assumed not physically connected between clusters). The remaining steps show how to indicate that a private network is shared between clusters.

3. Using the `mmchconfig` command for the CL2 cluster, add the `subnets='10.200.0.0/CL2.kgn.ibm.com;CL3.kgn.ibm.com'` attribute. Alternatively, regular expressions are allowed here, such as `subnets='10.200.0.0/CL[23].kgn.ibm.com'`. See note “[2](#)” on page 519 for the syntax allowed for the regular expressions.

This attribute indicates that the private 10.200.0.0 network extends to all nodes in clusters CL2 or CL3. This way, any two nodes in the CL2 and CL3 clusters can communicate through the Gigabit Ethernet.

This setting allows all CL2 nodes to communicate over their Gigabit Ethernet. Matching `CL3.kgn.ibm.com` with the cluster list for 10.200.0.0 allows remote mounts between clusters CL2 and CL3 to communicate over their Gigabit Ethernet.

4. Using the `mmchconfig` command for the CL3 cluster, add the `subnets='10.200.0.0/CL3.kgn.ibm.com;CL2.kgn.ibm.com'` attribute, alternatively `subnets='10.200.0.0/CL[32].kgn.ibm.com'`.

This attribute indicates that the private 10.200.0.0 network extends to all nodes in clusters CL2 or CL3. This way, any two nodes in the CL2 and CL3 clusters can communicate through the Gigabit Ethernet.

Matching of `CL3.kgn.ibm.com` with the cluster list for 10.200.0.0 allows all CL3 nodes to communicate over their Gigabit Ethernet, and matching `CL2.kgn.ibm.com` with that list allows remote mounts between clusters CL3 and CL2 to communicate over their Gigabit Ethernet.

Use the `subnets` attribute of the `mmchconfig` command when you wish the GPFS cluster to leverage additional, higher performance network connections that are available to the nodes in the cluster, or between clusters.

Notes:

1. IBM Storage Scale 5.1.5 introduces the Multi-Rail Over TCP (MROT) feature that enables the use of multiple subnets. With MROT, the `subnets` attribute can be used to establish fault tolerance or automatic failover. All the interfaces corresponding to the IP addresses in the list are used. If some of them are down, GPFS will try to use the others for communication. In versions which do not have MROT implemented, use of the `subnets` attribute does not ensure a highly available system. If the GPFS daemon is using the IP address specified by the `subnets` attribute, and that interface goes down, GPFS does not switch to the other network. You can use the **mmdiag --network** command option to verify that the subnet is being used.

2. Each subnet can be listed at most once in each cluster. For example, specifying:

```
subnets='10.200.0.0/CL2.kgn.ibm.com 10.200.0.0/CL3.kgn.ibm.com'
```

where the 10.200.0.0 subnet is listed twice, is not allowed. Therefore, subnets that span multiple clusters have to be assigned a cluster name pattern or a semicolon-separated cluster name list. It is possible to combine these, for example, items in semicolon-separated cluster lists can be plain names or regular expressions, as in the following:

```
subnets='1.0.0.1/CL[23].kgn.ibm.com;OC.xyz.ibm.com'
```

The following shows examples of patterns that are accepted:

```
[af3] matches letters 'a' and 'f', and number 3  
[0-7] matches numbers 0, 1, ... 7  
[a-p0-7] matches letter a, b, ... p and numbers from 0 to 7 inclusive  
* matches any sequence of characters  
? matches any (one) character
```

With MROT when the `subnets` attribute lists multiple subnets, and if any of these multiple subnets are in both the local cluster and a specified remote cluster, then all the common subnets are used for communication. In versions which do not have MROT implemented, then only the first subnet that is common in the list is used for communication between the local and remote clusters.

As an example, suppose that the `subnets` attribute is set as follows, on cluster CL2.kgn.ibm.com:

```
subnets='10.200.0.0/CL[23].kgn.ibm.com 10.201.0.0/CL[23].kgn.ibm.com'
```

With MROT, if node CL2N1 on cluster CL2.kgn.ibm.com has network interfaces with IP addresses 10.200.0.1 and 10.201.0.1, and node CLN31 on cluster CL3.kgn.ibm.com has network interfaces with IP addresses 10.200.0.5 and 10.201.0.5 then the communication between these two nodes will flow over the 10.200.0.0 and 10.201.0.0 subnets. The CL2N1 node uses the network interfaces with IP address 10.200.0.1 and 10.201.0.1. The node CLN31 uses the network interfaces with IP address 10.200.0.5 and 10.201.0.5.

In versions which do not have MROT implemented, the communication between these two nodes will flow over the 10.200.0.0 subnet, with CL2N1 using the interface with IP address 10.200.0.1, and CLN31 using the interface with IP address 10.200.0.5.

Specifying a cluster name or a cluster name pattern for each subnet is only needed when a private network is shared across clusters. If the use of a private network is confined within the local cluster, then no cluster name is required in the subnet specification.

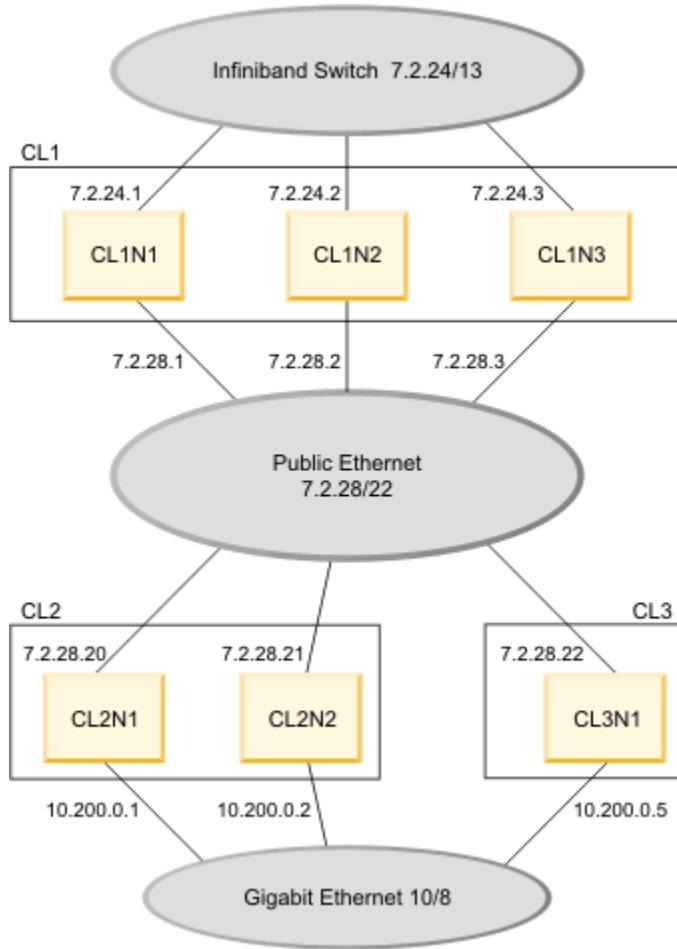


Figure 18. Use of public and private IP addresses in three GPFS clusters

Using multiple security levels for remote access

A cluster that owns a file system whose access is to be permitted from other clusters, can designate a different security level for each connecting cluster.

When multiple security levels are specified, each connection must use the security level of the connecting node unless that security level is AUTHONLY. In this case, the security level of the node that accepts the connection is used instead. This means that a connection must use AUTHONLY if both nodes exist in clusters that are required to use the AUTHONLY security method.

To specify a different security level for different clusters that request access to a specified cluster, use the `mmauth -l cipherList` command. The following examples illustrate:

1. In this example, `cluster1` and `cluster2` are on the same trusted network, and `cluster3` is connected to both of them with an untrusted network. The system administrator chooses these security levels:

- A `cipherList` of AUTHONLY for connections between `cluster1` and `cluster2`
- A `cipherList` of AES128-SHA for connections between `cluster1` and `cluster3`
- A `cipherList` of AES128-SHA for connections between `cluster2` and `cluster3`

The administrator of `cluster1` issues these commands:

```
mmauth add cluster2 -k keyFile -l AUTHONLY
mmauth add cluster3 -k keyFile -l AES128-SHA
```

2. In this example, `cluster2` is accessing file systems that are owned by `cluster1` by using a *cipherList* of AUTHONLY, but the administrator of `cluster1` decides to require a more secure *cipherList*. The administrator of `cluster1` issues this command:

```
mmauth update cluster2 -l AES128-SHA
```

Existing connections is upgraded from AUTHONLY to AES128-SHA.

Changing security keys with remote access

When working with GPFS file systems accessed by other GPFS clusters, it might be necessary to generate a new public/private access key. This can be done without disturbing existing connections, provided the following procedure is followed.

To accomplish this, the cluster that owns and serves the file system is made to temporarily have two access keys (referred to as the 'old key' and the 'new key'), which are both valid at the same time. The clusters currently accessing the file system can then change from the old key to the new key without interruption of file system access.

In this example, `cluster1` is the name of the cluster that owns and serves a file system, and `cluster2` is the name of the cluster that has already obtained access to this file system, and is currently using it. Here, the system administrator of `cluster1` changes the access key without severing the connection obtained by `cluster2`.

1. On `cluster1`, the system administrator issues the `mmauth genkey new` command to generate a new public/private access key pair. The key pair is placed in `/var/mmfs/ssl`:

```
mmauth genkey new
```

After this command is issued, `cluster1` will have two keys (referred to as the 'old key' and the 'new key') that can both be used to access `cluster1` file systems.

2. The system administrator of `cluster1` now gives the file `/var/mmfs/ssl/id_rsa.pub` (that contains the new key) to the system administrator of `cluster2`, who desires to continue to access the `cluster1` file systems. This operation requires the two administrators to coordinate their activities, and must occur outside of the GPFS command environment.
3. On `cluster2`, the system administrator issues the `mmremotecluster update` command to make the new key known to his system:

```
mmremotecluster update cluster1 -k cluster1_id_rsa.pub
```

where:

`cluster1`

Is the real name of `cluster1` as given by the `mmlscluster` command on a node in `cluster1`.

`cluster1_id_rsa.pub`

Is the name of the file obtained from the administrator of `cluster1` in Step [“2” on page 521](#).

This permits the cluster desiring to mount the file system to continue mounting file systems owned by `cluster1`.

4. On `cluster1`, the system administrator verifies that all clusters desiring to access `cluster1` file systems have received the new key and activated it using the `mmremotecluster update` command.
5. On `cluster1`, the system administrator issues the `mmauth genkey commit` command to commit the new key as the only valid access key. The old key will no longer be accepted once this command completes successfully:

```
mmauth genkey commit
```

Once the new public key has been committed, the old public key will no longer be accepted. As a result, any remote cluster administrator who has not been given the new key (see the preceding Step

“2” on page 521) and run `mmremotecluster update` (see the preceding Step “3” on page 521) will no longer be able to mount file systems owned by `cluster1`.

Similarly, the administrator of `cluster2` might decide to change the access key for `cluster2`:

1. On `cluster2`, the system administrator issues the `mmauth genkey new` command to generate a new public/private access key pair. The key pair is placed in `/var/mmfs/ssl`:

```
mmauth genkey new
```

After this command is issued, `cluster2` will have two keys (referred to as the ‘old key’ and the ‘new key’) that can both be used when a connection is established to any of the nodes in `cluster2`.

2. The system administrator of `cluster2` now gives the file `/var/mmfs/ssl/id_rsa.pub` (that contains the new key) to the system administrator of `cluster1`, the owner of the file systems. This operation requires the two administrators to coordinate their activities, and must occur outside of the GPFS command environment.
3. On `cluster1`, the system administrator issues the `mmauth update` command to make the new key known to his system:

```
mmauth update cluster2 -k cluster2_id_rsa.pub
```

where:

`cluster2`

Is the real name of `cluster2` as given by the `mmlscluster` command on a node in `cluster2`.

`cluster2_id_rsa.pub`

Is the name of the file obtained from the administrator of `cluster2` in Step “2” on page 522.

This permits the cluster desiring to mount the file system to continue mounting file systems owned by `cluster1`.

4. The system administrator of `cluster2` verifies that the administrator of `cluster1` has received the new key and activated it using the `mmauth update` command.
5. On `cluster2`, the system administrator issues the `mmauth genkey commit` command to commit the new key as the only valid access key. The old key will no longer be accepted once this command completes successfully:

```
mmauth genkey commit
```

NIST compliance

The `nistCompliance` configuration variable allows the system administrator to restrict the set of available algorithms and key lengths to a subset of those approved by NIST.

About this task

The `nistCompliance` variable applies to security transport (tscomm security, key retrieval) only, not to encryption, which always uses NIST-compliant mechanisms.

For the valid values for `nistCompliance`, see `mmchconfig` command in the *IBM Storage Scale: Command and Programming Reference Guide*.

The `nistCompliance` configuration variable has been introduced on version 4.1. Clusters created prior to that release operate with the equivalent of that variable being set to `off`. Similarly, clusters created on prior versions and which are migrated to 4.1 will have `nistCompliance` set to `off`.

Remote Mounts and version 3.5 clusters

A cluster created on version 4.1 or higher, and operating with `nistCompliance` set to SP800-131A, will be unable to remote-mount a file system from a version 3.5 cluster, since the 4.1 cluster will not accept

the key from the latter, which is not NIST SP800-131A-compliant. To allow the version 4.1 cluster to remote-mount the version 3.5 cluster, issue the

```
mmchconfig nistCompliance=off
```

command on the version 4.1 cluster, before the `mmremotecluster add` command can be issued. The key exchange will work even if the version 4.1 cluster already has a NIST-compliant key.

Updating a cluster to nistCompliance SP800-131A

A cluster upgraded from prior versions may have the `nistCompliance` set to `off` and may be operating with keys which are not NIST SP800-131A-compliant. To upgrade the cluster to operate in NIST SP800-131A mode, the following procedure should be followed:

From a node in the cluster which is running version 4.1 or later, issue:

```
mmauth genkey new  
mmauth genkey commit
```

If remote clusters are present, follow the procedure described in the “[Changing security keys with remote access](#)” on page 521 section (under Chapter 38, “Accessing a remote GPFS file system,” on page 505) to update the key on the remote clusters.

Once all nodes in the cluster are running at least version 4.1, run the following command from one of the nodes in the cluster:

```
mmchconfig release=LATEST
```

From one of the nodes in the cluster, run the following command:

```
mmchconfig nistCompliance=SP800-131A
```

For clusters at the version 5.1 level or higher, setting `nistCompliance` to `off` is not allowed. The `nistCompliance` value must be set to SP800-131A. The existing clusters that are running with `nistCompliance` value set to `off` must be changed to SP800-131A before migrating the cluster to the version 5.1 level.

If you want to set the `nistCompliance` value to `off` or continue to upgrade the version 5.1 level or higher with `nistCompliance` value set to `off`, use the option `--accept-no-compliance-to-nist-standards`. For more information, see the topic *Completing the upgrade to a new level of IBM Storage Scale* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Note: It is not recommended to use the `--accept-no-compliance-to-nist-standards` option and this option might not be available in the subsequent releases.

Important information about remote access

There is some additional information about this topic that you should take into consideration.

When working with GPFS file systems accessed by nodes that belong to other GPFS clusters, consider the following points:

1. A file system is administered only by the cluster where the file system was created. Other clusters may be allowed to mount the file system, but their administrators cannot add or delete disks, change characteristics of the file system, enable or disable quotas, run the `mmfsck` command, and so forth. The only commands that other clusters can issue are list type commands, such as: `mmlsfs`, `mmlsdisk`, `mmlsmount`, and `mmdf`.
2. Since each cluster is managed independently, there is no automatic coordination and propagation of changes between clusters, like there is between the nodes within a cluster.

This means that if the administrator of `cluster1` (the owner of file system `gpfs1`) decides to delete it or rename it, the information for `gpfs1` in `cluster2` becomes obsolete, and an attempt to mount `gpfs1` from `cluster2` will fail. It is assumed that when such changes take place, the two

administrators will inform each other. The administrator of `cluster2` can then use the update or delete options of the `mmremoteefs` command to make the appropriate changes.

3. If the names of the contact nodes change, the name of the cluster changes, or the public key file changes, use the update option of the `mmremotecluster` command to reflect the changes.
4. Use the show option of the `mmremotecluster` and `mmremoteefs` commands to display the current information about remote clusters and file systems.
5. If the cluster that owns a file system has a `maxblocksize` configuration parameter that is different from the `maxblocksize` configuration parameter of the cluster that desires to mount a file system, a mismatch may occur and file system mount requests may fail with messages to this effect. Check your `maxblocksize` configuration parameters on both clusters using the `mmlsconfig` command. Correct any discrepancies with the `mmchconfig` command.
6. Before taking steps to enable the remote cluster mount, you must ensure that the root administrator of the remote cluster is fully trusted by the home cluster's root administrator. The administrator of the remote cluster must also fully trust the root administrator of the home cluster.

Chapter 39. Information lifecycle management for IBM Storage Scale

IBM Storage Scale can help you achieve information lifecycle management (ILM) efficiencies through powerful policy-driven automated tiered storage management. With the ILM toolkit, you can manage sets of files and pools of storage, and you can automate the management of file data.

Using these tools, GPFS can automatically determine where to physically store your data regardless of its placement in the logical directory structure. Storage pools, filesets and user-defined policies provide the ability to match the cost of your storage resources to the value of your data.

Note: Available on all IBM Storage Scale editions.

GPFS policy-based ILM tools allow you to:

- Create *storage pools* to provide a way to partition a file system's storage into collections of disks or a redundant array of independent disks (RAIDs) with similar properties that are managed together as a group. GPFS has three types of storage pools:
 - A required system storage pool that you create and manage through GPFS
 - Optional user storage pools that you create and manage through GPFS
 - Optional external storage pools that you define with GPFS policy rules and manage through an external application such as IBM Storage Protect
- Create *filesets* to provide a way to partition the file system namespace to allow administrative operations at a finer granularity than that of the entire file system. See “[Filesets](#)” on page 582.
- Create *policy rules* based on data attributes to determine initial file data placement and manage file data placement throughout the life of the file. See “[Policies for automating file management](#)” on page 531.

To work with ILM in the GUI, click **Files > Information Lifecycle**.

Use the following information to create and manage information lifecycle management policies in IBM Storage Scale:

Storage pools

Physically, a *storage pool* is a collection of disks or RAID arrays. Storage pools also allow you to group multiple storage systems within a file system.

Using storage pools, you can create tiers of storage by grouping storage devices based on performance, locality, or reliability characteristics. For example, one pool could be an enterprise class storage system that hosts high-performance Fibre Channel disks and another pool might consist of numerous disk controllers that host a large set of economical SATA disks.

There are two types of storage pools in GPFS, internal storage pools and external storage pools. Internal storage pools are managed within GPFS. External storage pools are managed by an external application such as IBM Storage Protect. For external storage pools, GPFS provides tools that allow you to define an interface that your external storage manager uses to access your data. GPFS does not manage the data placed in external storage pools. Instead, GPFS manages the movement of data to and from external storage pools. Storage pools allow you to perform complex operations such as moving, mirroring, or deleting files across multiple storage devices, providing storage virtualization and a single management context.

Internal GPFS storage pools are meant for managing online storage resources. External storage pools are intended for use as near-line storage and for archival and backup operations. However, both types of storage pools provide you with a method to partition file system storage for considerations such as:

- Improved price-performance by matching the cost of storage to the value of the data

- Improved performance by:
 - Reducing the contention for premium storage
 - Reducing the impact of slower devices
 - Allowing you to retrieve archived data when needed
- Improved reliability by providing for:
 - Replication based on need
 - Better failure containment
 - Creation of new storage pools as needed

For more information, see the following subtopics on internal storage pools and external storage pools:

Internal storage pools

Internal GPFS storage pools are controlled by GPFS policies and commands. There are two types of internal GPFS storage pools, the required system storage pool and up to seven optional user storage pools. The system storage pool contains metadata for each file and may also contain user data. User storage pools can only contain user data.

The internal GPFS storage pool to which a disk belongs is specified as an attribute of the disk in the GPFS cluster. You specify the disk attributes as a field in each disk descriptor when you create the file system or when adding disks to an existing file system. GPFS allows a maximum of eight internal storage pools per file system. One of these storage pools is the required system storage pool. The other seven internal storage pools are optional user storage pools.

GPFS assigns file data to internal storage pools under these circumstances:

- When the file is initially created; the storage pool is determined by the file placement policy that is in effect when at the time of file creation.
- When the attributes of the file, such as file size or access time, match the rules of a policy that directs GPFS to migrate the data to a different storage pool.

For additional information, refer to:

- [“The system storage pool” on page 526](#)
- [“The system.log storage pool” on page 527](#)
- [“User storage pools” on page 579](#)
- [“Managing storage pools” on page 527](#)

Related concepts

[External storage pools](#)

External pools provide storage space that is not directly connected to or managed by IBM Storage Scale.

The system storage pool

The system storage pool contains file system control structures, reserved files, directories, symbolic links, and special devices. It also contains the metadata that is associated with regular files, including indirect blocks, extended attributes, and other file information.

The system storage pool can also contain user data. Only one system storage pool exists in a file system, and it is automatically created when the file system is created.

Important: It is a good practice to use highly reliable disks and replication for the system storage pool because it contains system metadata.

The amount of metadata grows as you add files to the system. Therefore, it is a good practice to monitor the system storage pool to ensure that it contains enough unused space to accommodate growth.

The system storage pool typically requires a small percentage of the total storage capacity that GPFS manages. However, the percentage that is required by the system storage pool varies depending on the environment. You can monitor the amount of space that is available in the system storage pool with the

`mmdf` command. If the available space in the system storage pool runs low, you can increase the available space by purging files or adding disks to the system storage pool.

The system.log storage pool

By default the file system recovery log is stored in the system storage pool with file system metadata. The file system recovery log can also be placed in a dedicated pool that is called the `system.log` pool.

This storage pool must be created explicitly. It is highly recommended to only use storage that is as fast or even faster than what is used for the system storage pool. This recommendation is because of the high number of small synchronous data updates made to the recovery log. The block size for the `system.log` pool must be the same as the block size of the system pool.

The file system recovery log will only be stored in one pool.

The `system.log` storage pool is an optional dedicated storage pool that contains only the file system recovery logs. If you define this pool, then IBM Storage Scale uses it for all the file system recovery logs of the file system. Otherwise, the file system recovery logs are kept in the system storage pool. It is a good practice for the `system.log` pool to consist of storage media that is as fast as or faster than the storage media of the system storage pool. If the storage is nonvolatile, this pool can be used for the high-availability write cache (HAWC).

Managing storage pools

Managing your storage pools includes the following tasks:

Creating storage pools

The storage pool to which a disk belongs is an attribute of each disk and is specified as a field in each disk descriptor when the file system is created using the `mmcrfs` command or when disks are added to an existing file system with the `mmadddisk` command. Adding a disk with a new storage pool name in the disk descriptor automatically creates the storage pool.

Storage pool names:

- Must be unique within a file system, but not across file systems.
- Cannot be longer than 255 alphanumeric characters.
- Are case sensitive. `MYpool` and `myPool` are distinct storage pools.

A storage pool is defined by the stanza keyword `pool`; for example:

```
pool=dataPoolA
```

If a storage pool is not specified, the disk is by default assigned to the system storage pool.

The `metadata-block-size` flag on the `mmcrfs` command can be used to create a system pool with a different block size from the user pools. This can be especially beneficial if the default block size is larger than 1 MB. If data and metadata block sizes differ, the system pool must contain only `metadataOnly` disks. For more information, see the topic *Block size* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Changing the storage pool assignment of a disk

Once a disk is assigned to a storage pool, the pool assignment cannot be changed by using either the `mmchdisk` command or the `mmrpldisk` command. You can, however, change the pool to which the disk is assigned.

To move a disk to another pool:

1. Delete the disk from its current pool by issuing the `mmdeldisk` command. This will move the data to the remaining disks in the storage pool.
2. Add the disk to the new pool by issuing the `mmadddisk` command.

3. Rebalance the data across all disks in the new storage pool by issuing the `mmrestripefs -P` command.

Changing the storage pool assignment of a file

You can change the storage pool that a file is assigned to.

A root user can change the storage pool that a file is assigned to by either:

- Running `mmapplypolicy` with an appropriate set of policy rules.
- Issuing the `mmchattr -P` command.

By default, both of these commands migrate data immediately (this is the same as using the `-I yes` option for these commands). If desired, you can delay migrating the data by specifying the `-I defer` option for either command. Using the `defer` option, the existing data does not get moved to the new storage pool until either the `mmrestripefs` command or the `mmrestripefile` command are executed. For additional information, refer to:

- [“Overview of policies” on page 531](#)
- [“Rebalancing files in a storage pool” on page 529](#)

Deleting storage pools

The system storage pool, the `system.log` pool, and user storage pools have different deletion requirements.

Deleting the system storage pool is not allowed. You can delete the system storage pool only after you have deleted the file system.

You can delete the `system.log` pool by deleting all the disks in the `system.log` pool. You do not need to run a policy to empty the `system.log` pool first, because the `system.log` pool can only contain log files, and those are automatically migrated to the System pool when you delete the `system.log` pool.

In order to delete a user storage pool, you must delete all its disks using the `mmdeledisk` command. When GPFS deletes the last remaining disk from a user storage pool, the storage pool is also deleted. To delete a storage pool, it must be completely empty. A migration policy along with the `mmapplypolicy` command could be used to do this.

Listing the storage pools of a file system

To list the storage pools available for a specific file system, issue the `mmlsfs -P` command.

For example, this command:

```
mmlsfs fs1 -P
```

produces output similar to this:

flag	value	description
<code>-P</code>	<code>system;sp1;sp2</code>	Disk storage pools in file system

For file system `fs1`, there are three storage pools: the system storage pool and user storage pools named `sp1` and `sp2`.

Listing the storage pool of a file

To display the assigned storage pool and the name of the fileset that includes the file, issue the `mmlsattr -L` command.

For example, this command:

```
mmlsattr -L myfile
```

produces output similar to this:

```
file name: myfile
metadata replication: 2 max 2
```

```

data replication:      1 max 2
immutable:            no
appendOnly:           no
flags:
storage pool name:   sp1
fileset name:         root
snapshot name:
creation Time:       Wed Feb 22 15:16:29 2012
Misc attributes:      ARCHIVE

```

File myfile is assigned to the storage pool named sp1 and is part of the root fileset.

Listing disks and associated statistics

To list the disks belonging to a storage pool, issue the `mmdf -P` command.

For example, this command:

```
mmdf fs1 -P sp1
```

produces output similar to this:

disk name	disk size in KB	failure group	holds metadata	holds data	free KB in full blocks	free KB in fragments
<hr/>						
Disks in storage pool: sp1 (Maximum disk size allowed is 1.2 TB)						
vp4vsdn05	17760256	6	no	yes	11310080 (64%)	205200 (1%)
vp5vsdn05	17760256	6	no	yes	11311104 (64%)	205136 (1%)
vp6vsdn05	17760256	6	no	yes	11300352 (64%)	206816 (1%)
vp7vsdn05	17760256	6	no	yes	11296256 (64%)	209872 (1%)
vp0vsdn05	17760256	6	no	yes	11293696 (64%)	207968 (1%)
vp1vsdn05	17760256	6	no	yes	11293184 (64%)	206464 (1%)
vp2vsdn05	17760256	6	no	yes	11309056 (64%)	203248 (1%)
vp3vsdn05	17760256	6	no	yes	11269120 (63%)	211456 (1%)
(pool total)	142082048				90382848 (64%)	1656160 (1%)

This example shows that storage pool sp1 in file system fs1 consists of eight disks and identifies details for each disk including:

- Name
- Size
- Failure group
- Data type
- Free space

Rebalancing files in a storage pool

A root user can rebalance file data across all disks in a file system by issuing the `mmrestripefs` command.

Optionally:

- Specifying the `-P` option rebalances only those files assigned to the specified storage pool.
- Specifying the `-p` option rebalances the file placement within the storage pool. For files that are assigned to one storage pool, but that have data in a different pool, (referred to as ill-placed files), using this option migrates their data to the correct pool. (A file becomes "ill-placed" when the `-I defer` option is used during migration of the file between pools.)

Using replication in a storage pool

To enable data replication in a storage pool, you must make certain that there are at least two failure groups within the storage pool.

This is necessary because GPFS maintains separation between storage pools and performs file replication within each storage pool. In other words, a file and its replica must be in the same storage pool. This also

means that if you are going to replicate the entire file system, every storage pool in the file system must have at least two failure groups.

Note: Depending on the configuration of your file system, if you try to enable file replication in a storage pool having only one failure group, GPFS will either give you a warning or an error message.

External storage pools

External pools provide storage space that is not directly connected to or managed by IBM Storage Scale.

External pools in IBM Storage Scale can be represented by a variety of tools that include IBM Spectrum Protect for Space Management, IBM Storage Scale Transparent Cloud Tiering, and IBM Spectrum Archive Enterprise Edition (EE). These tools allow files from the IBM Storage Scale file system to migrate to another storage system that is not directly connected to and managed by IBM Storage Scale.

External storage pools use a flexible interface driven by GPFS policy rules that simplify data migration to and from other types of storage such as tape storage. For more information, refer to [“Policies for automating file management” on page 531](#).

You can define multiple external storage pools at any time using GPFS policy rules. To move data to an external storage pool, the GPFS policy engine evaluates the rules that determine which files qualify for transfer to the external pool. From that information, GPFS provides a list of candidate files and executes the script specified in the rule that defines the external pool. That executable script is the interface to the external application, such as IBM Storage Protect, that does the actual migration of data into an external pool. Using the external pool interface, GPFS gives you the ability to manage information by allowing you to:

1. Move files and their extended attributes onto low-cost near-line or offline storage when demand for the files diminishes.
2. Recall the files, with all of their previous access information, onto online storage whenever the files are needed.

External pool requirements

With external pools, GPFS provides metadata processing and the flexibility of using extended file attributes. The external storage manager is responsible for moving files from GPFS and returning them upon the request of an application accessing the file system. Therefore, when you are using external storage pools, you must use an external file management application such as IBM Storage Protect. The external application is responsible for maintaining the file once it has left the GPFS file system. For example, GPFS policy rules create a list of files that are eligible for migration. GPFS hands that list to IBM Storage Protect which migrates the files to tape and creates a reference file in the file system that has pointers to the tape image. When a file is requested, it is automatically retrieved from the external storage pool and placed back in an internal storage pool. As an alternative, you can use a GPFS policy rule to retrieve the data in advance of a user request.

The number of external storage pools is only limited by the capabilities of your external application. GPFS allows you to define external storage pools at any time by writing a policy that defines the pool and makes that location known to GPFS. External storage pools are defined by policy rules and initiated by either storage thresholds or use of the `mapplypolicy` command.

For more information, refer to [“Working with external storage pools” on page 570](#).

Related concepts

[Internal storage pools](#)

Internal GPFS storage pools are controlled by GPFS policies and commands. There are two types of internal GPFS storage pools, the required system storage pool and up to seven optional user storage

pools. The system storage pool contains metadata for each file and may also contain user data. User storage pools can only contain user data.

Policies for automating file management

GPFS provides a means to automate the management of files using policies and rules. Properly managing your files allows you to efficiently use and balance your premium and less expensive storage resources.

GPFS supports the following policies:

- *File placement policies* are used to automatically place newly created files in a specific storage pool.
- Snapshot placement policies are used to automatically place snapshot data in a specific storage pool.
- *File management policies* are used to manage files during their lifecycle by moving them to another storage pool, moving them to near-line storage, copying them to archival storage, changing their replication status, or deleting them. They can also be used to migrate snapshot data to other storage pools or change its replication status.
- *Transparent Cloud Tiering policies* are used to migrate cold data to a cloud storage tier or recall data from the cloud storage tier on reaching certain threshold levels.

You can create and manage policies and policy rules with both the command line interface and the GUI. In the GUI, navigate to **Files > Information Lifecycle Management**.

Overview of policies

A *policy* is a set of rules that describes the life cycle of user data based on the attributes of files. Each rule defines an operation or definition, such as "migrate to a pool and replicate the file."

You can perform the following tasks with rules:

- Initial file placement
- Snapshot data placement
- File management
- Restoring file data
- Encryption-specific uses. For more information, see the topic *Encryption in the IBM Storage Scale: Command and Programming Reference Guide*.
- File compression and decompression. For more information about file compression and decompression, see the topics [“Policy rules: Terms” on page 535](#) and [“File compression” on page 231](#).

When a file is created or restored, the placement policy determines the location of the file's data and assigns the file to a storage pool. All data written to that file is placed in the assigned storage pool.

Similarly, if the file system has snapshots and a file is written to, the snapshot placement policy determines the storage pool where the snapshot blocks are placed.

The placement policy defining the initial placement of newly created files, snapshot data, and the rules for placement of restored data must be installed into GPFS with the `mmchpolicy` command. If a GPFS file system does not have a placement policy installed, all the data is stored in the first data storage pool. Only one placement policy can be installed at a time. If you switch from one placement policy to another, or make changes to a placement policy, that action has no effect on existing files. However, newly created files are always placed according to the currently installed placement policy.

The management policy determines file management operations such as migration, deletion, and file compression or decompression.

In order to migrate or delete data, you must use the `mmapplypolicy` command. To compress or decompress data, you can use either the `mmapplypolicy` command with a MIGRATE rule or the `mmchattr` command. You can define the file management rules and install them in the file system together with the placement rules. As an alternative, you may define these rules in a separate file and explicitly provide them to `mmapplypolicy` using the -P option. In either case, policy rules for placement

or migration may be intermixed. Over the life of the file, data can be migrated to a different storage pool any number of times, and files can be deleted or restored.

Note: In a multicluster environment, the scope of the `mmapplypolicy` command is limited to the nodes in the cluster that owns the file system.

Note: File compression or decompression using the `mmapplypolicy` command is not supported on the Windows operating system.

File management rules can also be used to control the space utilization of GPFS online storage pools. When the utilization for an online pool exceeds the specified high threshold value, GPFS can be configured, through user exits, to trigger an event that can automatically start `mmapplypolicy` and reduce the utilization of the pool. Using the `mmaddcallback` command, you can specify a script that will run when such an event occurs. For more information, see the topic *mmaddcallback command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

GPFS performs error checking for file-placement policies in the following phases:

- When you install a new policy, GPFS checks the basic syntax of all the rules in the policy.
- GPFS also checks all references to storage pools. If a rule in the policy refers to a storage pool that does not exist, the policy is not installed and an error is returned.
- When a new file is created, the rules in the active policy are evaluated in order. If an error is detected, GPFS logs an error, skips all subsequent rules, and returns an `EINVAL` error code to the application.
- Otherwise, the first applicable rule is used to store the file data.

Default file placement policy:

When a GPFS file system is first created, the default file placement policy is to assign all files to the first data storage pool. You can go back to the default policy by running the command:

```
mmchpolicy Device DEFAULT
```

For more information on using GPFS commands to manage policies, see [“Managing policies” on page 563](#).

Related concepts

Policy rules

A *policy rule* is an SQL-like statement that tells GPFS what to do with the data for a file in a specific storage pool if the file meets specific criteria. A rule can apply to any file being created or only to files being created within a specific fileset or group of filesets.

The `mmapplypolicy` command and policy rules

The `mmapplypolicy` command has policy rules that are based on the characteristics of different phases.

Working with external storage pools

With external storage pools you can migrate files to storage pools managed by an external application such as IBM Storage Protect.

Backup and restore with storage pools

When you back up data or restore data to a storage pool, consider the following descriptions.

ILM for snapshots

ILM for snapshots can be used to migrate snapshot data.

Related tasks

Managing policies

Policies and the rules that they contain are used to assign files to specific storage pools.

Related reference

Policy rules: Examples and tips

Before you write and apply policies, consider the following advice.

Policy rules

A *policy rule* is an SQL-like statement that tells GPFS what to do with the data for a file in a specific storage pool if the file meets specific criteria. A rule can apply to any file being created or only to files being created within a specific fileset or group of filesets.

A policy rule specifies one or more conditions that, when true, cause the rule to be applied. Conditions can be specified by SQL expressions, which can include SQL functions, variables, and file attributes. Some of the many available file attributes are shown in the following list. For more information, see “[File attributes in SQL expressions](#)” on page 541:

- Date and time when the rule is evaluated, that is, the current date and time
- Date and time when the file was last accessed
- Date and time when the file was last modified
- Fileset name
- File name or extension
- File size
- User ID and group ID

Note: Some file attributes are not valid in all types of policy rules.

GPFS evaluates policy rules in order, from first to last, as they appear in the policy. The first rule that matches determines what is to be done with that file. For example, when a client creates a file, GPFS scans the list of rules in the active file placement policy to determine which rule applies to the file. When a rule applies to the file, GPFS stops processing the rules and assigns the file to the appropriate storage pool. If no rule applies, an EINVAL error code is returned to the application.

There are nine types of policy rules that allow you to define specific actions that GPFS will implement on the file data. Each rule has clauses that control candidate selection, namely when the rule is allowed to match a file, what files it will match, the order to operate on the matching files and additional attributes to show for each candidate file. Different clauses are permitted on different rules based upon the semantics of the rule.

Related concepts

[Overview of policies](#)

A *policy* is a set of rules that describes the life cycle of user data based on the attributes of files. Each rule defines an operation or definition, such as “migrate to a pool and replicate the file.”

[The mmapplypolicy command and policy rules](#)

The `mmapplypolicy` command has policy rules that are based on the characteristics of different phases.

[Working with external storage pools](#)

With external storage pools you can migrate files to storage pools managed by an external application such as IBM Storage Protect.

[Backup and restore with storage pools](#)

When you back up data or restore data to a storage pool, consider the following descriptions.

[ILM for snapshots](#)

ILM for snapshots can be used to migrate snapshot data.

Related tasks

[Managing policies](#)

Policies and the rules that they contain are used to assign files to specific storage pools.

Related reference

[Policy rules: Examples and tips](#)

Before you write and apply policies, consider the following advice.

Policy rules: Syntax

Policy rules can apply to file placements, snapshot placements, group pools, file migrations, file deletions, file exclusions, file lists, file restores, external storage pool definitions, and external list definitions.

The policy rules and their respective syntax diagrams are as follows. For more information about encryption-specific rules, see [Chapter 49, “Encryption,” on page 735](#).

- File placement rules

```
RULE ['RuleName']
  SET POOL 'PoolName'
  [LIMIT (OccupancyPercentage)]
  [REPLICATE (DataReplication)]
  [FOR FILESET ('FilesetName',['FilesetName'])...]
  [ACTION (SqlExpression)]
  [WHERE SqlExpression]
```

- Snapshot placement rule

```
RULE ['RuleName']
  SET SNAP_POOL 'PoolName'
  [LIMIT (OccupancyPercentage)]
  [REPLICATE (DataReplication)]
  [FOR FILESET ('FilesetName',['FilesetName'])...]
  [ACTION (SqlExpression)]
  [WHERE SqlExpression]
```

- Group pool rule; used to define a list of pools that may be used as a pseudo-pool source or destination in either a FROM POOL or TO POOL clause within another rule

```
RULE ['RuleName'] GROUP POOL ['groupPoolName']
  IS 'poolName' [LIMIT(OccupancyPercentage)]
  THEN 'poolName2' [LIMIT(n2)]
  THEN 'pool-C' [LIMIT(n3)]
  THEN ...
```

- File migration rule

```
RULE ['RuleName'] [WHEN TimeBooleanExpression]
  MIGRATE
  [COMPRESS ({'yes' | 'no' | 'z' | 'lz4' | 'zfast' | 'alphae' | 'alphah'})]
  [FROM POOL 'FromPoolName']
  [THRESHOLD (HighPercentage[,LowPercentage[,PremigratePercentage]]])
  [WEIGHT (WeightExpression)]
  TO POOL 'ToPoolName'
  [LIMIT (OccupancyPercentage)]
  [REPLICATE (DataReplication)]
  [FOR FILESET ('FilesetName',['FilesetName'])...]
  [SHOW ([String] SqlExpression)]
  [SIZE (numeric-sql-expression)]
  [ACTION (SqlExpression)]
  [WHERE SqlExpression]
```

For more information about file compression and decompression, see the topic [“Policy rules: Terms” on page 535](#) and the topic *File compression* in the *IBM Storage Scale: Administration Guide*.

- File deletion rule

```
RULE ['RuleName'] [WHEN TimeBooleanExpression]
  DELETE
  [DIRECTORIES_PLUS]
  [FROM POOL 'FromPoolName']
  [THRESHOLD (HighPercentage[,LowPercentage])]
  [WEIGHT (WeightExpression)]
  [FOR FILESET ('FilesetName',['FilesetName'])...]
  [SHOW ([String] SqlExpression)]
  [SIZE (numeric-sql-expression)]
  [ACTION (SqlExpression)]
  [WHERE SqlExpression]
```

- File exclusion rule

```

RULE ['RuleName'] [WHEN TimeBooleanExpression]
  EXCLUDE
    [DIRECTORIES_PLUS]
    [FROM POOL 'FromPoolName']
    [FOR FILESET ('FilesetName' [, 'FilesetName']...)]
    [ACTION (SqlExpression)]
    [WHERE SqlExpression]

```

- File list rule

```

RULE ['RuleName'] [WHEN TimeBooleanExpression]
  LIST 'ListName'
    [EXCLUDE]
    [DIRECTORIES_PLUS]
    [FROM POOL 'FromPoolName']
    [THRESHOLD (HighPercentage[, LowPercentage])]
    [WEIGHT (WeightExpression)]
    [FOR FILESET ('FilesetName' [, 'FilesetName']...)]
    [SHOW (['String'] SqlExpression)]
    [SIZE (numeric-sql-expression)]
    [ACTION (SqlExpression)]
    [WHERE SqlExpression]

```

- File restore rule

```

RULE ['RuleName']
  RESTORE TO POOL 'PoolName'
    [LIMIT (OccupancyPercentage)]
    [REPLICATE (DataReplication)]
    [FOR FILESET ('FilesetName' [, 'FilesetName']...)]
    [ACTION (SqlExpression)]
    [WHERE SqlExpression]

```

- External storage pool definition rule

```

RULE ['RuleName']
  EXTERNAL POOL 'PoolName'
    EXEC 'InterfaceScript'
      [OPTS 'OptionsString ...']
      [ESCAPE '%SpecialCharacters']
      [SIZE sum-number]

```

- External list definition rule

```

RULE ['RuleName']
  EXTERNAL LIST 'ListName'
    EXEC 'InterfaceScript'
      [OPTS 'OptionsString ...']
      [ESCAPE '%SpecialCharacters']
      [THRESHOLD 'ResourceClass']
      [SIZE sum-number]

```

Policy rules: Terms

The terms of policy rules specify conditions for selecting files and operations to perform on files.

The following terms are used in policy rules. Some terms appear in more than one rule:

ACTION (SqlExpression)

Specifies an SQL expression that is evaluated only if the other clauses of the rule are satisfied. The action of the *SqlExpression* is completed, and the resulting value of the *SqlExpression* is discarded. In the following example, the rule sets the extended attribute "user.action" to the value "set pool s6" for files that begin with the characters "sp". These files are assigned to the system pool:

```

rule 's6' set pool 'system' action(setxattr('user.action','set pool s6')) where name like
'sp%'

```

Note: Encryption policies do not support the ACTION clause.

[COMPRESS ({'yes' | 'no' | 'z' | 'lz4' | 'zfast' | 'alphae' | 'alphah'})]

Indicates that the selected files are to be compressed or decompressed. The compression libraries are intended primarily for the following uses:

z

Cold data. Favors compression efficiency over access speed.

lz4

Active, nonspecific data. Favors access speed over compression efficiency.

zfast

Active genomic data in FASTA, SAM, or VCF format.

alphae

Active genomic data in FASTQ format. Slightly favors compression efficiency over access speed.

alphah

Active genomic data in FASTQ format. Slightly favors access speed over compression efficiency.

The following table summarizes the effect of each option on uncompressed or compressed files:

Table 45. Effects of options on uncompressed or compressed files.

Option	Uncompressed files	Compressed files generated with a different compression library	Compressed files generated with the same compression library
yes	Compress with z	Not affected	Not affected
no	Not affected	Decompress	Decompress
z	Compress with z	Re-compress with z	Not affected
lz4	Compress with lz4	Re-compress with lz4	Not affected
zfast	Compress with zfast	Re-compress with zfast	Not affected
alphae	Compress with alphae	Re-compress with alphae	Not affected
alphah	Compress with alphah	Re-compress with alphah	Not affected

For more information, see the topic *File compression* in the *IBM Storage Scale: Administration Guide*.

Examples:

- The following rule compresses the files in the pool **datapool** that begin with the string **green%**. Because the policy term **COMPRESS** specifies **yes** instead of a compression library, compression is done with the default compression library, which is the z library.

```
RULE 'COMPR1' MIGRATE FROM POOL 'datapool' COMPRESS('yes') WHERE NAME LIKE 'green%'
```

- The following rule compresses genomic data in files with the extensions .fastq and .fq:

```
RULE 'COMPRESS_GENOMIC' MIGRATE COMPRESS('alphae') WHERE lower(NAME) LIKE '%.fastq' OR lower(NAME) LIKE '%.fq'
```

DIRECTORIES_PLUS

Indicates that non-regular file objects (directories, symbolic links, and other items) must be included in the list. If not specified, only ordinary data files are included in the candidate lists.

DELETE

Identifies a file deletion rule. A file that matches this rule becomes a candidate for deletion.

ESCAPE '%SpecialCharacters'

Specifies that path names and the SHOW('string') expressions within the associated file lists are encoded with a scheme based on RFC3986 URI percent encoding.

Compare the two following rules:

```
RULE 'xp' EXTERNAL POOL 'pool-name' EXEC 'script-name' ESCAPE '%'  
RULE 'x1' EXTERNAL LIST 'list-name' EXEC 'script-name' ESCAPE '%/+@#'
```

Both rules specify that all characters except the "unreserved" characters in the set a-zA-Z0-9-_~ are encoded as %XX, where XX comprises two hexadecimal digits.

However, the GPFS ESCAPE syntax adds to the set of "unreserved" characters. In the first rule, the syntax ESCAPE '%' specifies a rigorous RFC3986 encoding. Under this rule, a path name such as /root/directory/@abc+def#ghi.jkl appears in a file list in the following format:

```
%2Froot%2Fdirectory%2F%40abc%2Bdef%23ghi.jkl
```

In the second rule, the syntax ESCAPE '%/+@#' specifies that none of the characters in set /+@# are escaped. Under this rule, the same path name appears in a file list in the following format:

```
/root/directory/@abc+def#ghi.jkl
```

If you omit the ESCAPE clause, the newline character is escaped as '\n', and the backslash character is escaped as '\\'; all other characters are presented as is, without further encoding.

EXCLUDE

Identifies a file exclusion rule.

RULE 'x' EXCLUDE

A file that matches this form of the rule is excluded from further consideration by any MIGRATE or DELETE rules that follow.

RULE 'rule-name' LIST 'listname-y' EXCLUDE

A file that matches this form of the rule is excluded from further consideration by any LIST rules that name the same *listname-y*.

EXEC 'InterfaceScript'

Specifies an external program to be invoked to pass requests to an external storage management application. *InterfaceScript* must be a fully qualified path name to a user-provided script or program that supports the commands described in “User-provided program for managing external pools” on page 572.

EXTERNAL LIST *ListName*

Defines an external list. This rule does not match files. It provides the binding between the lists that are generated with regular LIST rules with a matching *ListName* and the external program that you want to run with these lists as input.

EXTERNAL POOL *PoolName*

Defines an external storage pool. This rule does not match files but defines the binding between the policy language and the external storage manager that implements the external storage.

FOR FILESET ('FilesetName','FilesetName')...

Specifies that the rule applies only to files within the specified filesets.

FROM POOL *FromPoolName*

Specifies the name of the source pool from which files are candidates for migration.

GROUP POOL *PoolName*

Defines a group pool. This rule supports the concept of distributing data files over several GPFS disk pools.

Optionally, a LIMIT, specified as an occupancy percentage, can be specified for each disk pool; if not specified, the limit defaults to 99%. The THEN keyword signifies that disk pools that are specified before a THEN keyword are preferred over disk pools that are specified after. When a pool that is defined by a GROUP POOL rule is the TO POOL target of a MIGRATE rule, the selected files are distributed among the disk pools that comprise the group pool. Files of highest weight are put into the most preferred disk pool up to the occupancy limit for that pool. If more files must be migrated, they are put into the second most preferred pool up to the occupancy limit for that pool. Again, files of highest weight are selected.

If you specify a file that is defined by a GROUP POOL rule in a FROM POOL clause, the clause matches any file in any of the disk pools in the group pool.

You can “repack” a group pool by WEIGHT. Migrate files of higher weight to preferred disk pools by specifying a group pool as both the source and the target of a MIGRATE rule.

```
rule 'grpdef' GROUP POOL 'gpool' IS 'ssd' LIMIT(90) THEN 'fast' LIMIT(85) THEN 'sata'  
rule 'repack' MIGRATE FROM POOL 'gpool' TO POOL 'gpool' WEIGHT(FILE_HEAT)
```

For more information see “[User storage pools](#)” on page 579.

LIMIT (*OccupancyPercentage*)

Limits the creation of data in a storage pool. GPFS does not migrate a file into a pool if doing so exceeds the occupancy percentage for the pool. If you do not specify an occupancy percentage for a pool, the default value is 99%. See “[Phase two: Choosing and scheduling files](#)” on page 555.

You can specify *OccupancyPercentage* as a floating point number, as in the following example:

```
RULE 'r' RESTORE to pool 'x' limit(8.9e1)
```

For testing or planning purposes, and when you use the mmapplypolicy command with the -I defer or -I test option, you can specify a LIMIT larger than 100%.

The limit clause does not apply when the target TO POOL is a GROUP POOL. The limits that are specified in the rule that defines the target GROUP POOL govern the action of the MIGRATE rule.

LIST *ListName*

Identifies a file list generation rule. A file can appear in multiple different file lists if it matches more than one list rule but can appear in a single list only once. *ListName* provides the binding to an EXTERNAL LIST rule that specifies the executable program to call when the generated list is processed.

MIGRATE

Identifies a file migration rule. A file that matches this rule becomes a candidate for migration to the pool specified by the TO POOL clause.

OPTS '*OptionsString*' ..!

Specifies optional parameters to be passed to the external program defined with the EXEC clause. *OptionsString* is not interpreted by the GPFS policy engine.

REPLICATE (*DataReplication*)

Overrides the default data replication factor. This value must be specified as 1, 2, or 3.

RESTORE TO POOL *PoolName*

Identifies a file restore rule. When you restore a file with the gpfs_fputattrswithpathname() subroutine, you can use this rule to match files against their saved attributes rather than the current file attributes. This rule also applies to a command that uses that subroutine, such as the IBM Storage Protect command dsmc restore.

RULE ['*RuleName*']

Initiates the rule statement. *RuleName* identifies the rule and is used in diagnostic messages.

SET POOL {*PoolName* | EXCLUDE}

Identifies an initial file placement rule.

PoolName

Specifies the name of the storage pool where all files that match the rule criteria are placed.

EXCLUDE

Specifies that files that match the rule criteria are excluded from further consideration and will not be stored in any pool. If you try to create a file that matches a SET POOL EXCLUDE rule, the file system API returns the error ENOSPC.

SET SNAP_POOL *PoolName*

Identifies an initial snapshot placement rule. *PoolName* specifies the name of the storage pool where all snapshot files that match the rule criteria are placed.

Note: The pool is only set when the file data is written to the snapshot, not when the snapshot is created.

SHOW ([*'String'*] *SqlExpression*)

Inserts the requested information (the character representation of the evaluated SQL expression *SqlExpression*) into the candidate list that is created by the rule when it deals with external storage pools. *String* is a literal value that gets echoed back.

This clause has no effect in matching files but can be used to define other attributes to be exported with the candidate file lists.

SIZE (*numeric-sql-expression*)

Is an optional clause of any MIGRATE, DELETE, or LIST rules that are used for choosing candidate files. *numeric-sql-expression* specifies the size of the file to be used when in calculating the total amount of data to be passed to a user script. The default is KB_ALLOCATED.

SIZE *sum-number*

Is an optional clause of the EXTERNAL_POOL and EXTERNAL_LIST rules. *sum-number* limits the total number of bytes in all of the files named in each list of files passed to your EXEC 'script'. If a single file is larger than *sum-number*, it is passed to your EXEC 'script' as the only entry in a "singleton" file list.

Specify *sum-number* as a numeric constant or a floating-point value.

Note: The value of *sum-number* is in kilobytes.

THRESHOLD (*HighPercentage*[,*LowPercentage*[,*PremigratePercentage*]])

Controls migration and deletion based on the percent of assigned pool storage that is occupied.

HighPercentage

Indicates that the rule is to be applied only if the occupancy percentage of the current pool of the file is greater than or equal to the *HighPercentage* value. Specify a nonnegative integer in the range 0-100.

LowPercentage

Indicates that MIGRATE and DELETE rules are to be applied until the occupancy percentage of the current pool of the file is reduced to less than or equal to the *LowPercentage* value. Specify a nonnegative integer in the range 0-100. The default is 0%.

PremigratePercentage

Defines an occupancy percentage of a storage pool that is below the lower limit. Files that lie between the lower limit *LowPercentage* and the pre-migrate limit *PremigratePercentage* are copied and become dual-resident in both the internal GPFS storage pool and the designated external storage pool. This option allows the system to free up space quickly by deleting pre-migrated files if the pool becomes full. Specify a nonnegative integer in the range 0 to *LowPercentage*. The default is the same value as *LowPercentage*.

Notes:

1. Percentage values can be specified as numeric constants or floating-point values.
2. This option applies only when you migrate to the external storage pool.
3. This option does not apply when the current rule operates on one group pool.

THRESHOLD (*ResourceClass*)

Specifies the type of capacity-managed resources that are associated with *ListName*. The following values are valid:

FILESET_QUOTAS

Indicates that the LIST rule must use the occupancy percentage of the "hard limit" fileset quota per the mm1quota and mmedquota commands.

FILESET_QUOTA_SOFT

Indicates that the LIST rule must use the occupancy percentage of the "soft limit" fileset quota per the mm1quota and mmedquota commands.

GROUP_QUOTAS

Indicates that the LIST rule must use the occupancy percentage of the "hard limit" group quota per the mmlsquota and mmedquota commands.

GROUP_QUOTA_SOFT

Indicates that the LIST rule must use the occupancy percentage of the "soft limit" group quota per the mmlsquota and mmedquota commands.

POOL_CAPACITIES

Indicates that the LIST rule uses the occupancy percentage of the pool when it applies the threshold rule. This value is the default value. This value is used if the threshold is not specified in the EXTERNAL_LIST rule but appears in the LIST rule.

USER_QUOTAS

Indicates that the LIST rule uses the occupancy percentage of the "hard limit" user quota per the mmlsquota and mmedquota commands.

USER_QUOTA_SOFT

Indicates that the LIST rule uses the occupancy percentage of the "soft limit" user quota per the mmlsquota and mmedquota commands.

Note: This option does not apply when the current rule operates on one group pool.

For more detail on how THRESHOLD can be used to control file migration and deletion, see “[Phase one: Selecting candidate files](#)” on page 553 and “[Pre-migrating files with external storage pools](#)” on page 575.

TO POOL *ToPoolName*

Specifies the name of the storage pool to which all the files that match the rule criteria are migrated. This phrase is optional if the COMPRESS keyword is specified.

WEIGHT (*WeightExpression*)

Establishes an order on the matching files. Specifies an SQL expression with a numeric value that can be converted to a double-precision floating point number. The expression can refer to any of the file attributes and can include any constants and any of the available SQL operators or built-in functions.

WHEN (*TimeBooleanExpression*)

Specifies an SQL expression that evaluates to TRUE or FALSE, depending only on the SQL built-in variable CURRENT_TIMESTAMP. If the WHEN clause is present and *TimeBooleanExpression* evaluates to FALSE, the rule is skipped.

The mmapplypolicy command assigns the CURRENT_TIMESTAMP when it begins processing. It uses either the actual Coordinated Universal Time date and time or the date specified with the -D option.

WHERE *SqlExpression*

Specifies an SQL expression that can reference file attributes as SQL variables, functions, and operators. Some attributes are not available to all rules. Compares the file attributes specified in the rule with the attributes of the file that is created.

SqlExpression must be an expression that evaluates to TRUE or FALSE, but can be any combination of standard SQL syntax expressions, including built-in functions.

Omitting the WHERE clause entirely is equivalent to writing WHERE TRUE. The WHERE clause must be the last clause of the rule.

SQL expressions for policy rules

A number of the available clauses in the GPFS policy rules utilize SQL expressions.

You can reference different file attributes as SQL variables and combine them with SQL functions and operators. Depending on the clause, the SQL expression must evaluate to either TRUE or FALSE, a numeric value, or a character string. Not all file attributes are available to all rules.

File attributes in SQL expressions

SQL expressions can include file attributes that specify certain clauses.

The following file attributes can be used in SQL expressions specified with the WHERE, WEIGHT, and SHOW clauses:

ACCESS_TIME

Specifies an SQL time stamp value for the date and time that the file was last accessed (POSIX atime).
See EXPIRATION_TIME.

BLOCKSIZE

Specifies the size, in bytes, of each block of the file.

CHANGE_TIME

Specifies an SQL time stamp value for the date and time that the file metadata was last changed (POSIX ctime).

CLONE_DEPTH

Specifies the depth of the clone tree for the file.

CLONE_IS_PARENT

Specifies whether the file is a clone parent.

CLONE_PARENT_FILESETID

Specifies the fileset ID of the clone parent. The fileset ID is available only if CLONE_PARENT_IS_SNAP is a nonzero value.

CLONE_PARENT_INODE

Specifies the inode number of the clone parent, or NULL if it is not a file clone.

CLONE_PARENT_IS_SNAP

Specifies whether the clone parent is in a snapshot.

CLONE_PARENT_SNAP_ID

Specifies the snapshot ID of the clone parent. The snapshot ID is available only if CLONE_PARENT_IS_SNAP is a nonzero value.

CREATION_TIME

Specifies an SQL time stamp value that is assigned when a file is created.

DEVICE_ID

Specifies the ID of the device that contains the directory entry.

DIRECTORY_HASH

Can be used to group files within the same directory.

DIRECTORY_HASH is a function that maps every PATH_NAME to a number. All files within the same directory are mapped to the same number and deeper paths are assigned to larger numbers.

DIRECTORY_HASH uses the following functions:

CountSubstr(*BigString*,*LittleString*)

Counts and returns the number of occurrences of *LittleString* in *BigString*.

HashToFloat(*StringValue*)

Is a hash function that returns a quasi-random floating point number ≥ 0 and < 1 , whose value depends on a string value. Although the result might appear random, HashToFloat(*StringValue*) always returns the same floating point value for a particular string value.

The following rule lists the directory hash values for three directories:

```
RULE 'y' LIST 'x1' SHOW(DIRECTORY_HASH)
LIST 'x1' /abc/tdir/randy1 SHOW(+3.49449638091027E+000)
LIST 'x1' /abc/tdir/ax      SHOW(+3.49449638091027E+000)
LIST 'x1' /abc/tdir/mmPolicy.8368.765871DF/mm_tmp/PWL.12 SHOW(+5.21282524359412E+000)
LIST 'x1' /abc/tdir/mmPolicy.31559.1E018912/mm_tmp/PWL.3 SHOW(+5.10672733094543E+000)
LIST 'x1' /abc/tdir/mmPolicy.31559.1E018912/mm_tmp/PWL.2 SHOW(+5.10672733094543E+000)
```

The following rule causes files within the same directory to be grouped and processed together during deletion. Grouping the files can improve the performance of GPFS directory-locking and caching.

```
RULE 'purge' DELETE WEIGHT(DIRECTORY_HASH) WHERE (deletion-criteria)
```

EXPIRATION_TIME

Specifies the expiration time of the file, expressed as an SQL time-stamp value. If the expiration time of a file is not set, its expiration time is SQL NULL. You can detect such files by checking for "EXPIRATION_TIME IS NULL".

Remember the following points:

- EXPIRATION_TIME is tracked independently from ACCESS_TIME and both values are maintained for immutable files.
- Expiration time and indefinite retention are independent attributes. You can change the value of either one without affecting the value of the other.

FILE_HEAT

Specifies the access temperature of a file based on the frequency of file access. With **FILE_HEAT** you can build policy rules that migrate "hotter", more frequently accessed files to faster tiers of storage and "cooler", less frequently accessed files to slower tiers of storage. For more information see "[File heat: Tracking file access temperature](#)" on page 579.

Note: The **FILE_HEAT** attribute is updated only when the **atime** attribute is updated. Be aware that the **-S** options of the **mmcrfs** command and the **mmcrfs** command control whether and when **atime** is updated. You can also override the **-S** option temporarily with mount options that are specific to IBM Storage Scale. For more information, see the topics *mmchfs command* and *mmcrfs command* in the *IBM Storage Scale: Command and Programming Reference Guide* and *atime values* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

FILE_SIZE

Specifies the current size or length of the file, in bytes.

FILESET_NAME

Specifies the fileset where the path name for the files is located, or is to be created.

Note: Using the FOR FILESET clause has the same effect and is more efficient to evaluate.

GENERATION

Specifies a number that is incremented whenever an INODE number is reused.

GROUP_ID

Specifies the numeric group ID of the file's group.

GROUP_NAME

Specifies the group name that is associated with GROUP_ID.

INODE

Specifies the file's inode number.

KB_ALLOCATED

Specifies the number of kilobytes of disk space allocated for the file data.

MODE

Displays the type and permission bits of a file as a 10-character string. The string has the same format as the first 10 characters of the output from the UNIX ls -l command. For example, -rwxr-xr-x is the MODE string of a file that can be read or executed by its owner, its group, or any user, but written only by its owner.

The first character of the MODE attributes displays the file type. The following values are supported:

d

Directory.

l

Link.

- c** Character device.
- b** Block device.
- p** Pipe.
- s** Socket.
- ?** Some other attribute, or unknown.

MISC_ATTRIBUTES

Specifies various miscellaneous file attributes. The value is a string of characters that are defined as follows:

- +** File access is controlled by an Access Control List (ACL).
- a** The file is appendOnly.
- A** Archive.
- c** The file is selected to be compressed.
- d** The data of the file is completely contained in the inode.
- D** Directory. To match all directories, you can use %D% as a wildcard character.
- e** Encrypted. A Microsoft Windows file attribute. Does not refer to IBM Storage Scale file encryption.
- E** The file has extended-attribute metadata.
- f** Some data blocks of the file are ill-placed with respect to the File Placement Optimizer (FPO) attributes of the file.
- F** Regular data file.
- H** Hidden. A Microsoft Windows file attribute.
- i** Not indexed by content. A Microsoft Windows file attribute.
- I** Some data blocks might be ill-placed.
- j** AFM append flag.
- J** Some data blocks might be ill-replicated.
- k** Remote attributes present. Internal to AFM.
- K** Some data blocks might be ill-compressed.
- L** Symbolic link.

- m** Empty directory.
- M** Co-managed.
- 2** Data blocks are replicated.
- o** Offline.
- 0** Other (not F, D, nor L). For example, a device or named pipe.
- p** Reparse point. A Microsoft Windows file attribute.
- P** Active File Management (AFM) summary flag. Indicates that at least one specific AFM flag is set: j, k, u, v, w, x, y, or z.
- r** Has streams. A Microsoft Windows file attribute.
- R** Read-only.
- s** Sparse. A Microsoft Windows file attribute.
- S** System. A Microsoft Windows file attribute.
- t** Temporary. A Microsoft Windows file attribute.
- u** File is cached. Internal to AFM.
- U** The file is trunc-managed.
- v** AFM create flag.
- V** Read-managed.
- w** AFM dirty data flag.
- W** Write-managed.
- x** AFM hard-linked flag.
- X** Immutability.
- y** AFM attribute-changed flag.
- Y** Indefinite retention.
- z** AFM local flag.
- Z** Secure deletion.

MODIFICATION_SNAPID

Specifies the integer ID of the snapshot after which the file was last changed. The value is normally derived with the SNAP_ID() built-in function that assigns integer values to GPFS snapshot names. This attribute allows policy rules to select files that are modified after a snapshot image is taken.

MODIFICATION_TIME

Specifies an SQL time stamp value for the date and time that the file data was last modified (POSIX mtime).

NAME

Specifies the name of a file.

NLINK

Specifies the number of hard links to the file.

PATH_NAME

Specifies a path for the file; the path includes the name of the file.

POOL_NAME

Specifies the storage pool where the file data is located.

Note: Using the FROM POOL clause has the same effect and is often preferable.

SNAP_NAME

Specifies the snapshot name that the snapshot file is part of.

Note: This attribute has an effect only when it is used in snapshot placement rules.

RDEVICE_ID

Specifies the device type for a device.

USER_ID

Specifies the numeric user ID of the owner of the file. To return the value of USER_ID when USER_NAME returns NULL, use COALESCE(USER_NAME, VARCHAR(USER_ID)).

USER_NAME

Specifies the user name that is associated with USER_ID.

Notes:

1. When file attributes are referenced in initial placement rules, only the following attributes are valid: CREATION_TIME, FILESET_NAME, GROUP_ID, MODE, NAME, SNAP_NAME, and USER_ID. The placement rules, like all rules with a clause, might also reference the current date and current time and use them to control matching.
2. When file attributes are used for restoring files, the attributes correspond to the attributes at the time of the backup, not to the current restored file.
3. For SQL expressions, if you want to show any of these attribute fields as strings (for example, FILE_HEAT), use SHOW(' [FILE_HEAT] ') rather than SHOW('FILE_HEAT'), as the latter is expanded.
4. All date attributes are evaluated in Coordinated Universal Time (a time standard abbreviated as UTC).
5. **Note:** To test whether a file is encrypted by IBM Storage Scale, do one of the following actions:

- In a policy, use the following condition:

```
XATTR('gpfs.Encryption') IS NOT NULL
```

- On the command line, issue the following command:

```
mmlsattr -L FileName
```

Using built-in functions

With GPFS, you can use built-in functions in comparison predicates, between predicates, in predicates, like predicates, mathematical value expressions, and boolean, string and numeric literals.

Extended attribute functions

You can use these functions to support access to the extended attributes of a file, and to support conversion of values to the supported SQL data types.

The following attribute functions can be used:

GetXattrs(pattern,prototype)

Returns extended attribute key=value pairs of a file for all extended attributes whose keys that match *pattern*. The key=value pairs are returned in the format specified by *prototype*.

If the value specified for *pattern* is '*' or empty then all keys are matched.

The *prototype* is a character string representing the format of a typical key=value pair. The *prototype* allows the user to specify which characters will be used to quote values, escape special code points, separate the key and value, and separate each key=value pair.

Some examples of the *prototype* argument include:

```
key~n=value^n,          # specify the escape characters
hexkey=hexvalue,        # specify either or both as hexadecimal values
"key\n"="value\n",       # specify quotes on either or both
key:"value^n";          # specify alternatives to = and ,
k:"v^n";                # allow key and value to be abbreviated
key,                   # specify keys only
"value~n";              # specify values only
key='value~n'&          # alternative quoting character
key=value               # do not use a ',' separator; use space instead
```

You may omit the last or both arguments. The defaults are effectively

`GetXattrs('*','key^n=hexvalue,')`.

The `GetXattrs` function returns an empty string for files that have no extended attributes with keys that match *pattern*.

The `GetXattrs` function is supported by the `mmapplypolicy` command, but it might return NULL in other contexts.

SetBGF(BlockGroupFactor)

Specifies how many file system blocks are laid out sequentially on disk to behave like a single large block. This option only works if `--allow-write-affinity` is set for the data pool. This applies only to a new data block layout; it does not migrate previously existing data blocks.

SetWAD(WriteAffinityDepth)

Specifies the allocation policy to be used. This option only works if `--allow-write-affinity` is set for the data pool. This applies only to a new data block layout; it does not migrate previously existing data blocks.

SetWADFG("WadfgValueString")

Indicates the range of nodes (in a shared nothing architecture) where replicas of blocks in the file are to be written. You use this parameter to determine the layout of a file in the cluster so as to optimize the typical access patterns of your applications. This applies only to a new data block layout; it does not migrate previously existing data blocks.

"*WadfgValueString*" is a semicolon-separated string identifying one or more failure groups in the following format:

```
FailureGroup1[;FailureGroup2[;FailureGroup3]]
```

where each *FailureGroupx* is a comma-separated string identifying the rack (or range of racks), location (or range of locations), and node (or range of nodes) of the failure group in the following format:

```
Rack1{:Rack2{...{:Rackx}}},Location1{:Location2{...{:Locationx}}},ExtLg1{:ExtLg2{...{:ExtLgx}}}
```

For example, the following value

```
1,1,1:2;2,1,1:2;2,0,3:4
```

means that the first failure group is on rack 1, location 1, extLg 1 or 2; the second failure group is on rack 2, location 1, extLg 1 or 2; and the third failure group is on rack 2, location 0, extLg 3 or 4.

If the end part of a failure group string is missing, it is interpreted as 0. For example, the following are interpreted the same way:

```
2  
2,0  
2,0,0
```

Notes:

1. Only the end part of a failure group string can be left off. The missing end part may be the third field only, or it may be both the second and third fields; however, if the third field is provided, the second field must also be provided. The first field must *always* be provided. In other words, every comma must both follow and precede a number; therefore, *none* of the following are valid:

```
2,0,  
2,  
,0,0  
0,,0  
,,0
```

2. Wildcard characters (*) are supported in these fields.

Here is an example of using setBGF, setWAD, and setWADFG:

```
RULE 'bgf' SET POOL 'pool1' WHERE NAME LIKE '%' AND setBGF(128) AND setWAD(1) AND  
setWADFG(1,0,1;2,0,1;3,0,1)
```

This rule has the same effect as the following command:

```
mmchattr --block-group-factor 128 --write-affinity-depth 1 --write-affinity-failuregroup "1,0,1;2,0,1;3,0,1" test
```

After installing this policy, a newly created file will have the same values for these three extended attributes as it would if mmchattr were used to set them:

```
(06:29:11) hs22n42:/sncfs # mmattrs -L test  
file name: test  
metadata replication: 3 max 3  
data replication: 3 max 3  
immutable: no  
appendOnly: no  
flags:  
storage pool name: system  
fileset name: root  
snapshot name:  
Block group factor: 128  
gpfs.BGF  
Write affinity depth: 1  
gpfs.WAD  
Write Affinity Depth Failure Group(FG) Map for copy:1 1,0,1  
gpfs.WADFG  
Write Affinity Depth Failure Group(FG) Map for copy:2 2,0,1  
Write Affinity Depth Failure Group(FG) Map for copy:3 3,0,1  
creation time: Sat Jun 8 06:28:50 2013  
Misc attributes: ARCHIVE
```

SetXattr('ExtendedAttributeName', 'ExtendedAttributeValue')

This function sets the value of the specified extended attribute of a file.

Successful evaluation of SetXattr in a policy rule returns the value TRUE and sets the named extended attribute to the specified value for the file that is the subject or object of the rule. This function is effective for policy rules (like MIGRATE and LIST) that are evaluated by mmapplypolicy and for the policy placement rule, SET POOL, when a data file is about to be created.

XATTR(*extended-attribute-name* [, *start* [, *length*]])

Returns the value of a substring of the extended attribute that is named by its argument as an SQL VARCHAR value, where:

extended-attribute-name

Specifies any SQL expression that evaluates to a character string value. If the named extended attribute does not exist, XATTR returns the special SQL value NULL.

Note: In SQL, the expression `NULL || AnyValue` yields NULL. In fact, with a few exceptions, the special SQL value of NULL "propagates" throughout an SQL expression, to yield NULL. A notable exception is that `(expression) IS NULL` always yields either TRUE or FALSE, never NULL.

For example, if you wish to display a string like `_NULL_` when the value of the extended attribute of a file is NULL you will need to code your policy rules file like this:

```
define(DISPLAY_NULL,[COALESCE($1,'_NULL_')])
rule external list 'a' exec ''
rule list 'a' SHOW(DISPLAY_NULL(xattr('user.marc'))||' and '||DISPLAY_NULL(xattr('user.eric')))
```

Here is an example execution, where either or both of the values of the two named extended attributes may be NULL:

```
mmapapplypolicy /gig/sill -P /ghome/makaplan/policies/display-null.policy -I test -L 2
...
WEIGHT(inf) LIST 'a' /gg/s11/cc SHOW(_NULL_ and _NULL_)
WEIGHT(inf) LIST 'a' /gg/s11/mm SHOW(yes-marc and _NULL_)
WEIGHT(inf) LIST 'a' /gg/s11/bb SHOW(_NULL_ and yes-eric)
WEIGHT(inf) LIST 'a' /gg/s11/tt SHOW(yes-marc and yes-eric)
```

GPFS/Policy/SQL is a subset of standard ISO/ANSI SQL, with additional extensions and modifications to facilitate GPFS/ILM. Regarding the NULL value, GPFS/Policy/SQL supports the "unknown value" meaning of NULL.

start

Is the optional starting position within the extended attribute value. The default is 1.

length

Is the optional length, in bytes, of the extended attribute value to return. The default is the number of bytes from the start to the end of the extended attribute string.

Note: `XATTR(name,i,k) == SUBSTR(XATTR(name),i,k)`.

Some extended attribute values represent numbers or timestamps as decimal or binary strings. Use the `TIMESTAMP`, `XATTR_FLOAT`, or `XATTR_INTEGER` function to convert extended attributes to SQL numeric or timestamp values:

XATTR_FLOAT(*extended-attribute-name* [, *start* [, *length*, [, *conversion_option*]]])

Returns the value of a substring of the extended attribute that is named by its argument, converted to an SQL double floating-point value, where:

extended-attribute-name

Specifies any SQL expression that evaluates to a character string value. If the named extended attribute does not exist, XATTR returns the special SQL value NULL.

start

Is the optional starting position within the extended attribute value. The default is 1.

length

Is the optional length, in bytes, of the extended attribute value to return. The default is the number of bytes from the start to the end of the extended attribute string. You can specify length as -1 to reach from the start to the end of the extended attribute string.

conversion_option

Specifies how the bytes are to be converted to a floating-point value. Supported options include:

- BIG_ENDIAN_DOUBLE or BD - a signed binary representation, IEEE floating, sign + 11 bit exponent + fraction. This is the default when executing on a "big endian" host OS, such as AIX on PowerPC®.
- BIG_ENDIAN_SINGLE or BS - IEEE floating, sign + 8-bit exponent + fraction.
- LITTLE_ENDIAN_DOUBLE or LD - bytewise reversed binary representation. This is the default when executing on a "little endian" host OS, such as Linux on Intel x86.
- LITTLE_ENDIAN_SINGLE or LS - bytewise-reversed binary representation.
- DECIMAL - the conventional SQL character string representation of a floating-point value.

Notes:

1. Any prefix of a conversion name can be specified instead of spelling out the whole name. The first match against the list of supported options is used; for example, L matches LITTLE_ENDIAN_DOUBLE.
2. If the extended attribute does not exist, the selected substring has a length of 0, or the selected bytes cannot be converted to a floating-point value, the function returns the special SQL value NULL.

XATTR_INTEGER(*extended-attribute-name* [, *start* [, *length*, [, *conversion_option*]]])

Returns the value of (a substring of) the extended attribute named by its argument, converted to a SQL LARGEINT value, where:

extended-attribute-name

Specifies any SQL expression that evaluates to a character string value. If the named extended attribute does not exist, XATTR returns the special SQL value NULL.

start

Is the optional starting position within the extended attribute value. The default is 1.

length

Is the optional length, in bytes, of the extended attribute value to return. The default is the number of bytes from the start to the end of the extended attribute string. You can specify length as -1 to reach from the start to the end of the extended attribute string.

conversion_option

Specifies how the bytes are to be converted to a LARGEINT value. Supported options include:

- BIG_ENDIAN - a signed binary representation, most significant byte first. This is the default when executing on a "big endian" host OS, such as AIX on PowerPC.
- LITTLE_ENDIAN - bytewise reversed binary representation. This is the default when executing on a "little endian" host OS, such as Linux on Intel x86.
- DECIMAL - the conventional SQL character string representation of an integer value.

Notes:

1. Any prefix of a conversion name can be specified instead of spelling out the whole name (B, L, or D, for example).
2. If the extended attribute does not exist, the selected substring has a length of 0, or the selected bytes cannot be converted to a LARGEINT value, the function returns the special SQL value NULL. For example:

```
XATTR_INTEGER('xyz.jim',5,-1,'DECIMAL')
```

String functions

You can use these string-manipulation functions on file names and literal values.

Important tips:

1. You must enclose strings in single-quotation marks.
2. You can include a single-quotation mark in a string by using two single-quotation marks. For example, 'a"b' represents the string a'b.

CHAR(expr[, length])

Returns a fixed-length character string representation of its *expr* argument, where:

expr

Can be any data type.

length

If present, must be a literal, integer value.

The resulting type is CHAR or VARCHAR, depending upon the particular function called.

The string that CHAR returns is padded with blanks to fill the *length* of the string. If *length* is not specified, it defaults to a value that depends on the type of the argument (*expr*).

Note: The maximum length of a CHAR (fixed length string) value is 255 bytes. The result of evaluating an SQL expression whose result is type CHAR may be truncated to this maximum length.

CONCAT(x,y)

Concatenates strings *x* and *y*.

HEX(x)

Converts an integer *x* into hexadecimal format.

LENGTH(x)

Determines the length of the data type of string *x*.

LOWER(x)

Converts string *x* into lowercase.

REGEX(String,'Pattern')

Returns TRUE if the pattern matches, FALSE if it does not. *Pattern* is a Posix extended regular expression.

Note: The policy SQL parser normally performs M4 macro preprocessing with square brackets set as the quote characters. Therefore, it is recommended that you add an extra set of square brackets around your REGEX pattern string; for example:

```
...WHERE REGEX(name,['^*[a-z]*$']) /* only accept lowercase alphabetic file names */
```

The following SQL expression:

```
NOT REGEX(STRING_VALUE,['^*[^z]*$|^*[^y]*$|^*[^x]*$|[abc] '])
```

can be used to test if STRING_VALUE contains *all* of the characters x, y, and z, in any order, and *none* of the characters a, b, or c.

REGEXREPLACE(string,pattern,result-prototype-string)

Returns a character string as *result-prototype-string* with occurrences of \i (where *i* is 0 through 9) replaced by the substrings of the original string that match the *i*th parenthesis delimited parts of the pattern string. For example:

```
REGEXREPLACE('speechless',['([aeiou]*)([aeiou]*)(.*)'], ['last=\3. middle=\2. first=\1.'])
```

returns the following:

```
'last=chless. middle=ee. first=sp.'
```

When *pattern* does not match *string*, REGEXREPLACE returns the value NULL.

When a \0 is specified in the *result-prototype-string*, it is replaced by the substring of *string* that matches the entire *pattern*.

SUBSTR(x,y,z)

Extracts a portion of string *x*, starting at position *y*, optionally for *z* characters (otherwise to the end of the string). This is the short form of SUBSTRING. If *y* is a negative number, the starting position is counted from the end of the string; for example, SUBSTR('ABCDEFGH', -3, 2) == 'FG'.

Note: Do not confuse SUBSTR with substr. substr is an m4 built-in macro function.

SUBSTRING(x FROM y FOR z)

Extracts a portion of string x, starting at position y, optionally for z characters (otherwise to the end of the string).

UPPER(x)

Converts the string x into uppercase.

VARCHAR(expr [, length])

Returns a varying-length character string representation of a character string, date/time value, or numeric value, where:

expr

Can be any data type.

length

If present, must be a literal, integer value.

The resulting type is CHAR or VARCHAR, depending upon the particular function called. Unlike CHAR, the string that the VARCHAR function returns is not padded with blanks.

Note: The maximum length of a VARCHAR(variable length string) value is 8192 bytes. The result of evaluating an SQL expression whose result is type VARCHAR may be truncated to this maximum length.

Numerical functions

You can use numeric-calculation functions to place files based on either numeric parts of the file name, numeric parts of the current date, or UNIX-client user IDs or group IDs.

These functions can be used in combination with comparison predicates and mathematical infix operators (such as addition, subtraction, multiplication, division, modulo division, and exponentiation).

INT(x)

Converts number x to a whole number, rounding up fractions of .5 or greater.

INTEGER(x)

Converts number x to a whole number, rounding up fractions of .5 or greater.

MOD(x,y)

Determines the value of x taken modulo y (x % y).

Date and time functions

You can use these date-manipulation and time-manipulation functions to place files based on when the files are created and the local time of the GPFS node serving the directory where the file is being created.

CURRENT_DATE

Determines the current date on the GPFS server.

CURRENT_TIMESTAMP

Determines the current date and time on the GPFS server.

DAYOFWEEK(x)

Determines the day of the week from date or timestamp x. The day of a week is from 1 to 7 (Sunday is 1).

DAYOFYEAR(x)

Determines the day of the year from date x. The day of a year is a number from 1 to 366.

DAY(x)

Determines the day of the month from date or timestamp x.

DAYS(x)

Determines the number of days between date or timestamp x and 0001-01-01.

DAYSINMONTH(x)

Determines the number of days in the month of date x.

DAYSINYEAR(x)

Determines the day of the year of date x.

HOUR(x)

Determines the hour of the day (a value from 0 to 23) of timestamp x.

MINUTE(x)

Determines the minute from timestamp x.

MONTH(x)

Determines the month of the year from date or timestamp x.

QUARTER(x)

Determines the quarter of year from date x. Quarter values are the numbers 1 through 4. For example, January, February, and March are in quarter 1.

SECOND(x)

Returns the seconds portion of timestamp x.

TIMESTAMP(sql-numeric-value) or TIMESTAMP(sql-character-string-value)

Accepts any numeric value. The numeric value is interpreted as the number of seconds since January 1, 1970 (the standard UNIX epoch) and is converted to an SQL TIMESTAMP value.

Signed 64-bit LARGEINT argument values are supported. Negative argument values cause TIMESTAMP to convert these values to timestamps that represent years before the UNIX epoch.

This function also accepts character strings of the form YYYY-MM-DD HH:MM:SS. A hyphen (-) or an at sign (@) might appear instead of the blank between the date and the time. The time can be omitted.

An omitted time defaults to 00:00:00. The :SS field can be omitted, which defaults to 00.

WEEK(x)

Determines the week of the year from date x.

YEAR(x)

Determines the year from date or timestamp x.

All date and time functions use Universal Time (UT).

Example of a policy rules file

```
/*
Sample GPFS policy rules file
*/

rule 'vip' set pool 'pool0' where USER_ID <= 100
RULE 'm1' SET POOL 'pool1' WHERE LOWER(NAME) LIKE '%marc%'
RULE SET POOL 'pool1' REPLICATE (2) WHERE UPPER(NAME) = '%IBM%'
RULE 'r2' SET POOL 'pool2' WHERE UPPER(SUBSTRING(NAME FROM 1 FOR 4)) = 'GPFS'
RULE 'r3' SET POOL 'pool3' WHERE LOWER(SUBSTR(NAME,1,5)) = 'roger'
RULE SET POOL 'pool4' WHERE LENGTH(NAME) = 7
RULE SET POOL 'pool5' WHERE name like 'xyz%' AND name like '%qed' OR name like '%.tmp%'
RULE SET POOL 'pool6' WHERE name like 'abc%' OR name like '%xyz' AND name like 'x%'

RULE 'restore' RESTORE TO POOL 'pool0' where USER_ID <= 100

/* If none of the rules matches put those files in system pool */
rule 'default' SET POOL 'system'
```

Miscellaneous SQL functions

The following miscellaneous SQL functions are available.

SNAP_ID(['FilesetName'], 'SnapshotName')

Given an (optional) fileset/inode-space name and a snapshot name, this function returns the numeric snapshot ID of the given snapshot within the given inode-space.

GetEnv('EnvironmentVariableName')

This function gets the value of the specified environment variable.

GetMMconfig('GPFSConfigurationParameter')

This function gets the value of the specified GPFS configuration parameter.

The mmapplypolicy command and policy rules

The `mmapplypolicy` command has policy rules that are based on the characteristics of different phases.

Any given file is a potential candidate for at most one MIGRATE or DELETE operation during each invocation of the `mmapplypolicy` command. A single invocation of the `mmapplypolicy` command is called the *job*.

The `mmapplypolicy` command sets the SQL built-in variable `CURRENT_TIMESTAMP`, and collects pool occupancy statistics at the beginning of the job.

The `mmapplypolicy` job consists of three major phases:

1. [“Phase one: Selecting candidate files” on page 553](#)
2. [“Phase two: Choosing and scheduling files” on page 555](#)
3. [“Phase three: Migrating and deleting files” on page 557](#)

Related concepts

[Overview of policies](#)

A *policy* is a set of rules that describes the life cycle of user data based on the attributes of files. Each rule defines an operation or definition, such as “migrate to a pool and replicate the file.”

[Policy rules](#)

A *policy rule* is an SQL-like statement that tells GPFS what to do with the data for a file in a specific storage pool if the file meets specific criteria. A rule can apply to any file being created or only to files being created within a specific fileset or group of filesets.

[Working with external storage pools](#)

With external storage pools you can migrate files to storage pools managed by an external application such as IBM Storage Protect.

[Backup and restore with storage pools](#)

When you back up data or restore data to a storage pool, consider the following descriptions.

[ILM for snapshots](#)

ILM for snapshots can be used to migrate snapshot data.

Related tasks

[Managing policies](#)

Policies and the rules that they contain are used to assign files to specific storage pools.

Related reference

[Policy rules: Examples and tips](#)

Before you write and apply policies, consider the following advice.

Phase one: Selecting candidate files

In the first phase of the `mmapplypolicy` job, all the files within the specified GPFS file system device, or below the input path name, are scanned. The attributes of each file are read from the file's GPFS inode structure.

Tip: The `mmapplypolicy` command always does Phase one, even if no file data has changed and even if the purpose for running the command is only to rewrap encryption keys. This process can take a long time and can involve considerable system resources if the affected file system or fileset is very large. You might want to delay running `mmapplypolicy` until a time when the system is not running a heavy load of applications.

Note: `mmapplypolicy` reads directly from the metadata disk blocks and can therefore lag behind the POSIX state of the file system. To be sure that `MODIFICATION_TIME` and the other timestamps are completely up to date, you can use the following suspend-and-resume sequence to force recent changes to disk:

```
mmfsctl fs-name suspend; mmfsctl fs-name resume;
```

For each file, the policy rules are considered, in order, from first rule to last:

- If the rule has a WHEN clause that evaluates to FALSE, the rule is skipped.
- If the rule has a FROM POOL clause, and the named pool does not match the POOL_NAME attribute of the file, the rule is skipped. A FROM POOL clause that specifies a group pool name matches a file if any pool name within the group pool matches the POOL_NAME attribute of the file.
- If there is a THRESHOLD clause and the current pool of the file has an occupancy percentage that is less than the *HighPercentage* parameter of the THRESHOLD clause, the rule is skipped.
- If the rule has a FOR FILESET clause, but none of the named filesets match the FILESET_NAME attribute of the file, the rule is skipped.
- If the rule has a WHERE clause that evaluates to FALSE, the rule is skipped. Otherwise, the rule applies.
- If the applicable rule is a LIST '*listname-y*' rule, the file becomes a candidate for inclusion in the named list unless the EXCLUDE keyword is present, in which case the file will not be a candidate; nor will any following LIST '*listname-y*' rules be considered for the subject file. However, the file is subject to LIST rules naming other list names.
- If the applicable rule is an EXCLUDE rule, the file will be neither migrated nor deleted. Files matching the EXCLUDE rule are not candidates for any MIGRATE or DELETE rule.

Note: Specify the EXCLUDE rule before any other rules that might match the files that are being excluded. For example:

```
RULE 'Exclude root's file' EXCLUDE where USER_ID = 0  
RULE 'Migrate all but root's files' MIGRATE TO POOL 'pool1'
```

will migrate all the files that are not owned by root. If the MIGRATE rule was placed in the policy file before the EXCLUDE rule, all files would be migrated because the policy engine would evaluate the rules from first to last, and root's files would have to match the MIGRATE rule.

To exclude files from matching a LIST rule, you must create a separate LIST rule with the EXCLUDE clause and place it before the LIST rule.

- If the applicable rule is a MIGRATE rule, the file becomes a *candidate* for migration to the pool specified by the TO POOL clause.

When a group pool is the TO POOL target of a MIGRATE rule, the selected files are distributed among the disk pools comprising the group pool, with files of highest weight going to the most preferred disk pool up to the occupancy limit for that pool. If there are still more files to be migrated, those go to the second most-preferred pool up to the occupancy limit for that pool (again choosing the highest-weight files from among the remaining selected files); and so on for the subsequent most-preferred pools, until either all selected files have been migrated or until all the disk pools of the group pool have been filled to their respective limits.

- If the applicable rule is a DELETE rule, the file becomes a *candidate* for deletion.
- If there is no applicable rule, the file is not a candidate for migration or deletion.
- Each candidate file (for migration or deletion) is also associated with a *LowPercentage* occupancy percentage value, which is taken from the THRESHOLD clause of the applicable rule. If not specified, the *LowPercentage* value defaults to 0%.
- Each candidate file is also associated with a numeric *weight*, either computed from the *WeightExpression* of the applicable rule, or assigned a default using these rules:
 - If a *LowPercentage* is specified within a THRESHOLD clause of the applicable rule, the weight of the candidate is taken as the KB_ALLOCATED attribute of the candidate file.
 - If a *LowPercentage* is not specified within a THRESHOLD clause of the applicable rule, the weight of the candidate is taken as +infinity.

Related concepts

[Phase two: Choosing and scheduling files](#)

In the second phase of the `mmapplypolicy` job, some or all of the candidate files are chosen.

Phase three: Migrating and deleting files

In the third phase of the `mmapplypolicy` job, the candidate files that were chosen and scheduled by the second phase are migrated or deleted, each according to its applicable rule.

Phase two: Choosing and scheduling files

In the second phase of the `mmapplypolicy` job, some or all of the candidate files are chosen.

Chosen files are scheduled for migration or deletion, taking into account the weights and thresholds determined in “[Phase one: Selecting candidate files](#)” on page 553, as well as the actual pool occupancy percentages. Generally, candidates with higher weights are chosen ahead of those with lower weights.

File migrations to and from external pools are done before migrations and deletions that involve only GPFS disk pools.

File migrations that do not target group pools are done before file migrations to group pools.

File migrations that target a group pool are done so that candidate files with higher weights are migrated to the more preferred GPFS disk pools within the group pool, but respecting the LIMITs specified in the group pool definition.

The following two options can be used to adjust the method by which candidates are chosen:

--choice-algorithm {best | exact | fast}

Specifies one of the following types of algorithms that the policy engine is to use when selecting candidate files:

best

Chooses the optimal method based on the rest of the input parameters.

If -B *MaxFiles* is used with this option, the number of files can be less than *MaxFiles*.

exact

Sorts all of the candidate files completely by weight, then serially considers each file from highest weight to lowest weight, choosing feasible candidates for migration, deletion, or listing according to any applicable rule LIMITs and current storage-pool occupancy. This is the default.

fast

Works together with the parallelized -g /shared-tmp -N node-list selection method. The fast choice method does not completely sort the candidates by weight. It uses a combination of statistical, heuristic, and parallel computing methods to favor higher weight candidate files over those of lower weight, but the set of chosen candidates may be somewhat different than those of the exact method, and the order in which the candidates are migrated, deleted, or listed is somewhat more random. The fast method uses statistics gathered during the policy evaluation phase. The fast choice method is especially fast when the collected statistics indicate that either all or none of the candidates are feasible.

If -B *MaxFiles* is used with this option, the number of files can be less than *MaxFiles*.

--split-margin *n.n*

A floating-point number that specifies the percentage within which the fast-choice algorithm is allowed to deviate from the LIMIT and THRESHOLD targets specified by the policy rules. For example if you specified a THRESHOLD number of 80% and a split-margin value of 0.2, the fast-choice algorithm could finish choosing files when it reached 80.2%, or it might choose files that bring the occupancy down to 79.8%. A nonzero value for split-margin can greatly accelerate the execution of the fast-choice algorithm when there are many small files. The default is 0.2.

File grouping and the SIZE clause

When scheduling files, `mmapplypolicy` simply groups together either the next 100 files by default, or the number of files explicitly set using the -B option.

However, you can set up `mmapplypolicy` to schedule files so that each invocation of the InterfaceScript gets approximately the same amount of file data to process. To do so, use the SIZE clause of certain

policy rules to specify that scheduling be based on the sum of the sizes of the files. The SIZE clause can be applied to the following rules (for details, see “[Policy rules](#)” on page 533):

- DELETE
- EXTERNAL LIST
- EXTERNAL POOL
- LIST
- MIGRATE

Administrator-specified customized file grouping or aggregation

In addition to using the SIZE clause to control the *amount* of work passed to each invocation of a InterfaceScript, you can also specify that files with *similar attributes* be grouped or aggregated together during the scheduling phase. To do so, use an aggregator program to take a list of chosen candidate files, sort them according to certain attributes, and produce a reordered file list that can be passed as input to the user script.

You can accomplish this by following these steps:

1. Run `mmapplypolicy` with the `-I` `prepare` option to produce a list of chosen candidate files, but not pass the list to a InterfaceScript.
2. Use your aggregator program to sort the list of chosen candidate files into groups with similar attributes and write each group to a new, separate file list.
3. Run `mmapplypolicy` with the `-r` option, specifying a set of file list files to be read. When invoked with the `-r` option, `mmapplypolicy` does not choose candidate files; rather, it passes the specified file lists as input to the InterfaceScript.

Note: You can also use the `-q` option to specify that small groups of files are to be taken in round-robin fashion from the input file lists (for example, take a small group of files from `x.list.A`, then from `x.list.B`, then from `x.list.C`, then back to `x.list.A`, and so on, until all of the files have been processed).

To prevent `mmapplypolicy` from redistributing the grouped files according to size, omit the SIZE clause from the appropriate policy rules and set the bunching parameter of the `-B` option to a very large value.

Reasons for candidates not to be chosen for deletion or migration

Generally, a candidate is not chosen for deletion from a pool, nor migration out of a pool, when the pool occupancy percentage falls below the *LowPercentage* value. Also, candidate files will not be chosen for migration into a target `T0_POOL` when the target pool reaches the occupancy percentage specified by the LIMIT clause (or 99% if no LIMIT was explicitly specified by the applicable rule).

The limit clause does not apply when the target `T0_POOL` is a group pool; the limits specified in the rule defining the target group pool govern the action of the MIGRATE rule. The policy-interpreting program (for example, `mmapplypolicy`) may issue a warning if a LIMIT clause appears in a rule whose target pool is a group pool.

Related concepts

[Phase one: Selecting candidate files](#)

In the first phase of the `mmapplypolicy` job, all the files within the specified GPFS file system device, or below the input path name, are scanned. The attributes of each file are read from the file's GPFS inode structure.

[Phase three: Migrating and deleting files](#)

In the third phase of the `mmapplypolicy` job, the candidate files that were chosen and scheduled by the second phase are migrated or deleted, each according to its applicable rule.

Phase three: Migrating and deleting files

In the third phase of the `mmapplypolicy` job, the candidate files that were chosen and scheduled by the second phase are migrated or deleted, each according to its applicable rule.

For migrations, if the applicable rule had a `REPLICATE` clause, the replication factors are also adjusted accordingly. It is acceptable for the effective `FROM POOL` and `TO POOL` to be the same because the `mmapplypolicy` command can be used to adjust the replication factors of files without necessarily moving them from one pool to another.

The migration performed in the third phase can involve large amounts of data movement. Therefore, you may want to consider using the `-I defer` option of the `mmapplypolicy` command, and then perform the data movements with the `mmrestripefs -p` command.

Related concepts

[Phase one: Selecting candidate files](#)

In the first phase of the `mmapplypolicy` job, all the files within the specified GPFS file system device, or below the input path name, are scanned. The attributes of each file are read from the file's GPFS inode structure.

[Phase two: Choosing and scheduling files](#)

In the second phase of the `mmapplypolicy` job, some or all of the candidate files are chosen.

Policy rules: Examples and tips

Before you write and apply policies, consider the following advice.

It is a good idea to test your policy rules by running the `mmapplypolicy` command with the `-I test` option and the `-L 3` or higher option. Testing helps you understand which files are selected as candidates and which candidates are chosen to be processed.

Do not apply a policy to an entire file system of important files unless you are confident that the rules correctly express your intentions. To test your rules, choose a directory that contains a small number of files, some of which you expect to be selected by your SQL policy rules and some of which you expect to be skipped.

Then enter a command like the following one:

```
mmapplypolicy /TestSubdirectory -L 6 -I test
```

The output shows which files are scanned and which match rules or no rules. If a problem is not apparent, you can add a `SHOW()` clause to your rule to see the values of file attributes or SQL expressions. To see multiple values, enter a command like the following one:

```
SHOW('x1=' || varchar(Expression1) || ' x2=' || varchar(Expression2) || ... )
```

where `ExpressionX` is the SQL variable or expression of function that you suspect or do not understand. Beware that if any expression evaluates to SQL `NULL`, then the entire show clause is `NULL`, by the rules of SQL. One way to show null vs. non-null values is to define a macro and call it as in the following example:

```
define(DISPLAY_NULL,[CASE WHEN ($1) IS NULL THEN '_NULL_' ELSE varchar($1) END])  
rule list a SHOW( 'x1=' || DISPLAY_NULL(xattr('user.marc')) || ' and x2=' ||  
DISPLAY_NULL(xattr('user.eric')))
```

Note: For examples and more information on the `-L` flag, see the topic *The mmapplypolicy -L command* in the *IBM Storage Scale: Problem Determination Guide*.

The following examples illustrate some useful policy rule techniques:

1. Delete files from the storage pool that is named pool_1 that were not accessed in the last 30 days and that are named like temporary files or appear in any directory that is named tmp:

```
RULE 'del1' DELETE FROM POOL 'pool_1'
    WHERE (DAYS(CURRENT_TIMESTAMP) - DAYS(ACCESS_TIME) > 30)
        AND (lower(NAME) LIKE '%.tmp' OR PATH_NAME LIKE '%/tmp/%')
```

2. Use the SQL LIKE predicate to test file names and path names:

```
RULE '*/_*'   DELETE WHERE PATH_NAME LIKE '%/x_%' ESCAPE 'x'
RULE '*XYZ*'  DELETE WHERE NAME LIKE '%XYZ%'
RULE '12_45'   DELETE WHERE NAME LIKE '12x_45' ESCAPE 'x'
RULE '12%45'   DELETE WHERE NAME LIKE '12x%45' ESCAPE 'x'
RULE '12?45'   DELETE WHERE NAME LIKE '12_45'
RULE '12*45'   DELETE WHERE NAME LIKE '12%45'
RULE '*_*'    DELETE WHERE NAME LIKE '%x_%' ESCAPE 'x'
```

Where:

- A percent % wildcard in the name represents zero or more characters.
- An underscore (_) wildcard in the name represents 1 byte.

Use the optional ESCAPE clause to establish an escape character, when you need to match '_' or '%' exactly.

3. Use the SQL UPPER and LOWER functions to ignore case when testing names:

```
RULE 'UPPER'  DELETE WHERE upper(PATH_NAME) LIKE '%/TMP/OLD/%'
RULE 'lower'  DELETE WHERE lower(PATH_NAME) LIKE '%/tmp/old/%'
```

4. Use the SQL SUBSTR or SUBSTRING functions to test a substring of a name:

```
RULE 's1' DELETE WHERE SUBSTRING(NAME FROM 1 FOR 5)='XXXX- '
RULE 's2' DELETE WHERE SUBSTR(NAME,1,5)='YYYY- '
```

5. Use the SQL SUBSTR and LENGTH functions to test the suffix of a name:

```
RULE 'sfx' DELETE WHERE SUBSTR(NAME,LENGTH(NAME)-3)='.tmp'
```

6. Use a WHEN clause to restrict rule applicability to a particular day of the week:

```
RULE 'D_SUN' WHEN (DayOfWeek(CURRENT_DATE)=1) /* Sunday */
    DELETE WHERE PATH_NAME LIKE '%/tmp/%'
```

CURRENT_DATE is an SQL built in operand that returns the date portion of the CURRENT_TIMESTAMP value.

7. Use the SQL IN operator to test several possibilities:

```
RULE 'D_WEEKEND' WHEN (DayOfWeek(CURRENT_DATE) IN (7,1)) /* Saturday or Sunday */
    DELETE WHERE PATH_NAME LIKE '%/tmp/%'
```

For information on how to use a macro processor such as m4 to make reading and writing policy rules easier, see “[Using macro processing utilities with policy rules](#)” on page 562.

8. Use a FILESET clause to restrict the rule to files within particular filesets:

```
RULE 'fsrule1' MIGRATE TO POOL 'pool_2'
    FOR FILESET('root','fset1')
```

In this example there is no FROM POOL clause, so regardless of their current storage pool placement, all files from the named filesets are subject to migration to storage pool pool_2.

Note: To have the migrate rule applied to snapshot files, you must specify the mmapplypolicy fs -S snap1 option, where snap1 is the name of the snapshot where the files reside.

9. Use an EXCLUDE rule to exclude a set of files from all subsequent rules:

```
RULE 'Xsuper' EXCLUDE WHERE USER_ID=0  
RULE 'mpg'     DELETE  WHERE lower(NAME) LIKE '%.mpg' AND FILE_SIZE>20123456
```

Notes:

- Specify the EXCLUDE rule before rules that might match the files that are being excluded.
- You cannot define a list and what to exclude from the list in a single rule. You must define two LIST statements, one specifying which files are in the list and one specifying what to exclude from the list. For example, to exclude files that contain the word test from the LIST rule allfiles, define the following rules:

```
RULE EXTERNAL LIST 'allfiles' EXEC '/u/brownap/policy/CHE/exec.list'  
  
RULE 'exclude_allfiles' LIST 'allfiles' EXCLUDE where name like '%test%'  
  
RULE 'all' LIST 'allfiles' SHOW('misc_attr ='|| MISC_ATTRIBUTES || HEX(MISC_ATTRIBUTES))  
\\ where name like '%'
```

10. Use the SQL NOT operator with keywords, along with AND and OR:

```
RULE 'D_WEEKDAY' WHEN (DayOfWeek(CURRENT_DATE) NOT IN (7,1)) /* a weekday */  
      DELETE WHERE (PATH_NAME LIKE '%/tmp/%' OR NAME LIKE '%.tmp')  
                 AND (KB_ALLOCATED > 9999 AND NOT USER_ID=0)
```

11. To migrate only snapshot files that for which data blocks are allocated, use the following rule:

```
RULE "migrate snap data" MIGRATE FROM POOL X TO POOL Y WHERE KB_ALLOCATED > 0
```

12. Use a REPLICATE clause to increase the availability of selected files:

```
RULE 'R2' MIGRATE FROM POOL 'ypooly' TO POOL 'ypooly'  
      REPLICATE(2) WHERE USER_ID=0
```

Before you increase the data replication factor for any file, the file system must be configured to support data replication.

13. The difference of two SQL Timestamp values can be compared to an SQL Interval value:

```
rule 'a' migrate to pool 'A' where CURRENT_TIMESTAMP - MODIFICATION_TIME > INTERVAL '10'  
  DAYS  
rule 'b' migrate to pool 'B' where CURRENT_TIMESTAMP - MODIFICATION_TIME > INTERVAL '10'  
  HOURS  
rule 'c' migrate to pool 'C' where CURRENT_TIMESTAMP - MODIFICATION_TIME > INTERVAL '10'  
  MINUTES  
rule 'd' migrate to pool 'D' where CURRENT_TIMESTAMP - MODIFICATION_TIME > INTERVAL '10'  
  SECONDS
```

For the best precision, use the INTERVAL...SECONDS construct.

14. By carefully assigning both weights and thresholds, the administrator can formally express rules like this:

If the storage pool named pool_X has an occupancy percentage greater than 90% now, bring the occupancy percentage of storage pool that is named pool_X down to 80% by migrating files that are three months or older to the storage pool that is named pool_ZZ. But, if you can find enough year-old files to bring the occupancy percentage down to 50%, do that also.

```
RULE 'year-old' MIGRATE FROM POOL 'pool_X'  
      THRESHOLD(90,50) WEIGHT(weight_expression)  
      TO POOL 'pool_ZZ'  
      WHERE DAYS(CURRENT_TIMESTAMP) - DAYS(ACCESS_TIME) > 365  
  
RULE '3month-old' MIGRATE FROM POOL 'pool_X'  
      THRESHOLD(90,80) WEIGHT(weight_expression)  
      TO POOL 'pool_ZZ'  
      WHERE DAYS(CURRENT_TIMESTAMP) - DAYS(ACCESS_TIME) > 90
```

More information about weights is available in the next example.

A goal of this `mmapplypolicy` job is to reduce the occupancy percentage of the `FROM POOL` to the low occupancy percentage specified on the `THRESHOLD` clause, if possible. The `mmapplypolicy` job does not migrate or delete more files than are necessary to produce this occupancy percentage. The task consists of these steps:

- a. Each candidate file is assigned a weight.
- b. All candidate files are sorted by weight.
- c. The highest weight files are chosen to `MIGRATE` or `DELETE` until the low occupancy percentage is achieved, or there are no more candidates.

The administrator who writes the rules must ensure that the computed weights are as intended, and that the comparisons are meaningful. This is similar to the IBM Storage Protect convention, where the weighting function for each file is determined by the equation:

```
X * access_age + Y * file_size
```

where:

`access_age` is `DAYS(CURRENT_TIMESTAMP) - DAYS(ACCESS_TIME)`
`file_size` is `FILE_SIZE` or `KB_ALLOCATED`

X and Y are weight factors that are chosen by the system administrator.

15. The `WEIGHT` clause can be used to express ideas like this (stated informally):

```
IF access_age > 365 days THEN weight = 100000 + access_age  
ELSE IF access_age < 30 days THEN weight = 0  
ELSE weight= KB_ALLOCATED
```

This rule means:

- Give a very large weight bias to any file older than a year.
- Force the weight of any file younger than 30 days to 0.
- Assign weights to all other files according to the number of kilobytes occupied.

The following code block shows the formal SQL syntax:

```
CASE  
WHEN DAYS(CURRENT_TIMESTAMP) - DAYS(ACCESS_TIME) > 365  
    THEN 100000 + DAYS(CURRENT_TIMESTAMP) - DAYS(ACCESS_TIME)  
WHEN DAYS(CURRENT_TIMESTAMP) - DAYS(ACCESS_TIME) < 30  
    THEN 0  
ELSE  
    KB_ALLOCATED  
END
```

16. The `SHOW` clause has no effect in matching files but can be used to define additional attributes to be exported with the candidate file lists. It can be used for any purpose but is primarily used to support file aggregation.

To support aggregation, you can use the `SHOW` clause to output an aggregation value for each file that is selected by a rule. You can then output those values to a file list and input that list to an external program that groups the files into aggregates.

17. If you have a large number of filesets against which to test, use the `FILESET_NAME` variable as shown in the following example:

```
RULE 'x' SET POOL 'gold' WHERE FILESET_NAME LIKE 'xyz%.xyz'
```

However, if you are testing against just a few filesets, you can use the `FOR FILESET('xyz1', 'xyz2')` form instead.

18. You can convert a time interval value to a number of seconds with the SQL cast syntax, as in the following example:

```

define([toSeconds],[((\$1) SECONDS(12,6))])
define([toUnixSeconds],[toSeconds(\$1 - '1970-1-1@0:00'))]
RULE external list b
RULE list b SHOW('sinceNow=' toSeconds(current_timestamp-modification_time) )
RULE external list c
RULE list c SHOW('sinceUnixEpoch=' toUnixSeconds(modification_time) )

```

The following method is also supported:

```

define(access_age_in_days,( INTEGER(( (CURRENT_TIMESTAMP - ACCESS_TIME) SECONDS)) /(24*3600.0) ) )
RULE external list w exec ''
RULE list w weight(access_age_in_days) show(access_age_in_days)

```

19. You can create a policy that lists the files that are created, accessed, or modified later than a specified timestamp. The timestamp must be converted from native format to UTC format. The following example policy lists all the files that were created after the timestamp 2017-02-21 04:56 IST:

```

cat policy
RULE 'filesRule' LIST 'files'
    SHOW(varchar(kb_allocated) || ' ' || varchar(file_size))
    WHERE (CREATION_TIME > TIMESTAMP('LAST_CREATE'))

```

To implement this policy, enter the following commands. The third line converts the time stamp to UTC format.

```

LC='2017-02-21 04:56 IST'
echo $LC
LCU=$(date +%Y-%m-%d" "%H:%M -d "$LC" -u)
echo $LCU
mmapplypolicy gpfs0 -P policy -I defer -f /tmp -M LAST_CREATE="$LCU"

```

The /tmp/list.files file contains the list of selected files.

You can modify the SHOW clause to list any file attribute. For more information, see [“File attributes in SQL expressions” on page 541](#).

20. To test whether a file is encrypted by IBM Storage Scale, use the following condition:

```
XATTR('gpfs.Encryption') IS NOT NULL
```

Related concepts

[Overview of policies](#)

A *policy* is a set of rules that describes the life cycle of user data based on the attributes of files. Each rule defines an operation or definition, such as "migrate to a pool and replicate the file."

[Policy rules](#)

A *policy rule* is an SQL-like statement that tells GPFS what to do with the data for a file in a specific storage pool if the file meets specific criteria. A rule can apply to any file being created or only to files being created within a specific fileset or group of filesets.

[The mmapplypolicy command and policy rules](#)

The mmapplypolicy command has policy rules that are based on the characteristics of different phases.

[Working with external storage pools](#)

With external storage pools you can migrate files to storage pools managed by an external application such as IBM Storage Protect.

[Backup and restore with storage pools](#)

When you back up data or restore data to a storage pool, consider the following descriptions.

[ILM for snapshots](#)

ILM for snapshots can be used to migrate snapshot data.

Related tasks

[Managing policies](#)

Policies and the rules that they contain are used to assign files to specific storage pools.

Using macro processing utilities with policy rules

Prior to evaluating the policy rules, GPFS invokes the m4 macro processor to process the policy file.

This processing allows you to incorporate into the policy file some of the traditional m4 facilities and to define simple and parameterized macros, conditionally include text, perform conditional evaluation, perform simple string operations, perform simple integer arithmetic and much more.

Note: GPFS uses the m4 built-in changequote macro to change the quote pair to [] and the changeocom macro to change the comment pair to /* */ (as in the C programming language).

Utilizing m4 as a front-end processor simplifies writing policies and produces policies that are easier to understand and maintain. Here is Example “15” on page 560 from “Policy rules: Examples and tips” on page 557 written with a few m4 style macro definitions:

```
define(access_age,(DAYS(CURRENT_TIMESTAMP) - DAYS(ACCESS_TIME)))

define(weight_expression,
CASE
WHEN access_age > 365
    THEN 100000 + access_age
WHEN access_age < 30
    THEN 0
ELSE
    KB_ALLOCATED
END
)

RULE year-old MIGRATE FROM POOL pool_X
    THRESHOLD(90,50) WEIGHT(weight_expression)
    TO POOL pool_ZZ
    WHERE access_age > 365

RULE 3month-old MIGRATE FROM POOL pool_X
    THRESHOLD(90,80) WEIGHT(weight_expression)
    TO POOL pool_ZZ
    WHERE access_age > 90
```

If you would like to use megabytes or gigabytes instead of kilobytes to represent file sizes, and SUNDAY, MONDAY, and so forth instead of 1, 2, and so forth to represent the days of the week, you can use macros and rules like this:

```
define(MB_ALLOCATED,(KB_ALLOCATED/1024.0))
define(GB_ALLOCATED,(KB_ALLOCATED/1048576.0))
define(SATURDAY,7)
define(SUNDAY,1)
define(MONDAY,2)
define(DAY_OF_WEEK, DayOfWeek(CURRENT_DATE))

RULE 'gb1' WHEN(DAY_OF_WEEK IN (SATURDAY,SUNDAY))
    MIGRATE TO POOL 'ypooly' WHERE GB_ALLOCATED >= .015

RULE 'mb4' MIGRATE TO POOL 'zpoolz' WHERE MB_ALLOCATED >= 4
```

The mmapplypolicy command provides a -M option that can be used to specify m4 macro definitions when the command is invoked. The policy rules may include variable identifiers whose values can be set using one or more -M options on the mmapplypolicy command. The policy rules could then compare file attributes to the currently provided values for the macro defined variables.

Among other things, this allows you to create a single policy file and reuse it for incremental backups without editing the file for each backup. For example, if your policy file contains the rules:

```
RULE EXTERNAL POOL 'archive' EXEC '/opts/hpss/archiveScript' OPTS '-server archive_server'
RULE 'mig1' MIGRATE TO POOL 'dead' WHERE ACCESS_TIME < TIMESTAMP(deadline)
RULE 'bak1' MIGRATE TO POOL 'archive' WHERE MODIFICATION_SNAPID > last_snapid
```

Then, if you invoke `mmapplypolicy` with these options:

```
mmapplypolicy ... -M "deadline='2006-11-30'" -M "last_snapid=SNAPID('2006_DEC')" \
-M archive_server="archive.abc.com"
```

The "mig1" rule will migrate old files that were not accessed since 2006/11/30 to an online pool named "dead". The "bak1" rule will migrate files that have changed since the 2006_DEC snapshot to an external pool named "archive". When the external script /opts/hpss/archiveScript is invoked, its arguments will include "-server archive.abc.com".

Managing policies

Policies and the rules that they contain are used to assign files to specific storage pools.

A storage pool typically contains a set of volumes that provide a specific quality of service for a specific use, such as to store all files for a particular application or a specific business division.

Managing policies includes:

- [“Creating a policy” on page 563](#)
- [“Installing a policy” on page 564](#)
- [“Changing the active policy” on page 565](#)
- [“Listing policies” on page 566](#)
- [“Validating policies” on page 566](#)
- [“Deleting policies” on page 567](#)
- [“Using thresholds to migrate data between pools” on page 567](#)
- [“Improving performance with the --sort-command parameter” on page 568](#)
- [“Improving performance in very large file systems” on page 569](#)

Related concepts

[Overview of policies](#)

A *policy* is a set of rules that describes the life cycle of user data based on the attributes of files. Each rule defines an operation or definition, such as "migrate to a pool and replicate the file."

[Policy rules](#)

A *policy rule* is an SQL-like statement that tells GPFS what to do with the data for a file in a specific storage pool if the file meets specific criteria. A rule can apply to any file being created or only to files being created within a specific fileset or group of filesets.

[The `mmapplypolicy` command and policy rules](#)

The `mmapplypolicy` command has policy rules that are based on the characteristics of different phases.

[Working with external storage pools](#)

With external storage pools you can migrate files to storage pools managed by an external application such as IBM Storage Protect.

[Backup and restore with storage pools](#)

When you back up data or restore data to a storage pool, consider the following descriptions.

[ILM for snapshots](#)

ILM for snapshots can be used to migrate snapshot data.

Related reference

[Policy rules: Examples and tips](#)

Before you write and apply policies, consider the following advice.

Creating a policy

Create a text file for your policy by following these guidelines.

- A policy must contain at least one rule.

- A policy file is limited to a size of 1 MB.
- When a file placement policy is applied to a file, the policy engine scans the list of rules in the policy in order, starting at the top, to determine which rule applies to the file. When the policy engine finds a rule that applies to the file, it stops processing the rules and assigns the file to the appropriate storage pool. If no rule applies, the policy engine returns an EINVAL error code to the application.

Note: The last placement rule of a policy rule list should be in the following form so that the file is assigned to a default pool if no other placement rule applies:

```
RULE 'DEFAULT' SET POOL 'default-data-pool'
```

For file systems that are upgraded to V4.1.1 or later: If there are no SET POOL policy rules installed to a file system by mmchpolicy, the system acts as if the single rule SET POOL '*first-data-pool*' is in effect, where *first-data-pool* is the firstmost non-system pool that is available for file data storage, if such a non-system pool is available. ("Firstmost" is the first according to an internal index of all pools.) However, if there are no policy rules installed and there is no non-system pool, the system acts as if SET POOL 'system' is in effect.

For file systems that are upgraded to V4.1.1: Until a file system is upgraded, if no SET POOL rules are present (set by mmchpolicy) for the file system, all data is stored in the 'system' pool.

- Comments within a policy must start with a /* and end with a */:

```
/* This is a comment */
```

For more information, see the topic [“Policy rules” on page 533](#).

Related concepts

[Using thresholds to migrate data between pools](#)

With the use of storage pools, the potential for a pool to exhaust its available space while other pools have free space can occur.

[Improving performance in very large file systems](#)

Read about how to improve the performance of the **mmapplypolicy** command in very large file systems.

Related tasks

[Installing a policy](#)

Install a policy by following these guidelines.

[Changing the active policy](#)

When you prepare a file with the new or changed policy rules, then issue the mmchpolicy command.

[Listing policies](#)

When you use the mmlspolicy command to list policies, follow these guidelines.

[Validating policies](#)

When you validate a policy file, follow this guideline.

[Deleting policies](#)

When you remove the current policy rules and restore the file-placement policy, follow this guideline.

[Improving performance with the --sort-command parameter](#)

To improve performance of the mmapplypolicy command, follow these guidelines.

Installing a policy

Install a policy by following these guidelines.

To install a policy:

1. Create a text file containing the desired policy rules.
2. Issue the mmchpolicy command.

Related concepts

[Using thresholds to migrate data between pools](#)

With the use of storage pools, the potential for a pool to exhaust its available space while other pools have free space can occur.

Improving performance in very large file systems

Read about how to improve the performance of the **mmapplypolicy** command in very large file systems.

Related tasks

Creating a policy

Create a text file for your policy by following these guidelines.

Changing the active policy

When you prepare a file with the new or changed policy rules, then issue the **mmchpolicy** command.

Listing policies

When you use the **mmlspolicy** command to list policies, follow these guidelines.

Validating policies

When you validate a policy file, follow this guideline.

Deleting policies

When you remove the current policy rules and restore the file-placement policy, follow this guideline.

Improving performance with the --sort-command parameter

To improve performance of the **mmapplypolicy** command, follow these guidelines.

Changing the active policy

When you prepare a file with the new or changed policy rules, then issue the **mmchpolicy** command.

The **mmchpolicy** command activates the following sequence of events:

1. The policy file is read into memory, and the information is passed to the current file system manager node.
2. The policy rules are validated by the file system manager.
3. If the policy file contains incorrect rules, no updates are made and an error is returned.
4. If no errors are detected, the new policy rules are installed in an internal file.

Policy changes take effect immediately on all nodes that have the affected file system mounted. For nodes that do not have the file system mounted, policy changes take effect upon the next mount of the file system.

Related concepts

Using thresholds to migrate data between pools

With the use of storage pools, the potential for a pool to exhaust its available space while other pools have free space can occur.

Improving performance in very large file systems

Read about how to improve the performance of the **mmapplypolicy** command in very large file systems.

Related tasks

Creating a policy

Create a text file for your policy by following these guidelines.

Installing a policy

Install a policy by following these guidelines.

Listing policies

When you use the **mmlspolicy** command to list policies, follow these guidelines.

Validating policies

When you validate a policy file, follow this guideline.

Deleting policies

When you remove the current policy rules and restore the file-placement policy, follow this guideline.

Improving performance with the --sort-command parameter

To improve performance of the **mmapplypolicy** command, follow these guidelines.

Listing policies

When you use the **mmlspolicy** command to list policies, follow these guidelines.

The **mmlspolicy** command displays policy information for a given file system. The information displayed is:

- When the policy file was installed.
- The user who installed the policy file.
- The first line of the original policy file.

The **mmlspolicy -L** command returns the installed (original) policy file. This shows all the rules and comments as they were in the policy file when it was installed. This is useful if you want to change policy rules - simply retrieve the original policy file using the **mmlspolicy -L** command and edit it.

Related concepts

[Using thresholds to migrate data between pools](#)

With the use of storage pools, the potential for a pool to exhaust its available space while other pools have free space can occur.

[Improving performance in very large file systems](#)

Read about how to improve the performance of the **mmapplypolicy** command in very large file systems.

Related tasks

[Creating a policy](#)

Create a text file for your policy by following these guidelines.

[Installing a policy](#)

Install a policy by following these guidelines.

[Changing the active policy](#)

When you prepare a file with the new or changed policy rules, then issue the **mmchpolicy** command.

[Validating policies](#)

When you validate a policy file, follow this guideline.

[Deleting policies](#)

When you remove the current policy rules and restore the file-placement policy, follow this guideline.

[Improving performance with the --sort-command parameter](#)

To improve performance of the **mmapplypolicy** command, follow these guidelines.

Validating policies

When you validate a policy file, follow this guideline.

The **mmchpolicy -I test** command validates but does *not* install a policy file.

Related concepts

[Using thresholds to migrate data between pools](#)

With the use of storage pools, the potential for a pool to exhaust its available space while other pools have free space can occur.

[Improving performance in very large file systems](#)

Read about how to improve the performance of the **mmapplypolicy** command in very large file systems.

Related tasks

[Creating a policy](#)

Create a text file for your policy by following these guidelines.

[Installing a policy](#)

Install a policy by following these guidelines.

[Changing the active policy](#)

When you prepare a file with the new or changed policy rules, then issue the `mmchpolicy` command.

[Listing policies](#)

When you use the `mmlspolicy` command to list policies, follow these guidelines.

[Deleting policies](#)

When you remove the current policy rules and restore the file-placement policy, follow this guideline.

[Improving performance with the --sort-command parameter](#)

To improve performance of the `mmapplypolicy` command, follow these guidelines.

Deleting policies

When you remove the current policy rules and restore the file-placement policy, follow this guideline.

To remove the current policy rules and restore the default GPFS file-placement policy, specify `DEFAULT` as the name of the policy file on the `mmchpolicy` command. This replaces the current policy file with default policy that assigns all newly created files to the first data storage pool.

Related concepts

[Using thresholds to migrate data between pools](#)

With the use of storage pools, the potential for a pool to exhaust its available space while other pools have free space can occur.

[Improving performance in very large file systems](#)

Read about how to improve the performance of the `mmapplypolicy` command in very large file systems.

Related tasks

[Creating a policy](#)

Create a text file for your policy by following these guidelines.

[Installing a policy](#)

Install a policy by following these guidelines.

[Changing the active policy](#)

When you prepare a file with the new or changed policy rules, then issue the `mmchpolicy` command.

[Listing policies](#)

When you use the `mmlspolicy` command to list policies, follow these guidelines.

[Validating policies](#)

When you validate a policy file, follow this guideline.

[Improving performance with the --sort-command parameter](#)

To improve performance of the `mmapplypolicy` command, follow these guidelines.

Using thresholds to migrate data between pools

With the use of storage pools, the potential for a pool to exhaust its available space while other pools have free space can occur.

To assist administrators in managing all available space in storage pools, and to mitigate the potential of a pool to become full, resulting in out of space errors on file system operations, the policy engine supports migration rules. Migration policies are implemented with the `MIGRATE` rule and are controlled via the `THRESHOLD` parameter to the `MIGRATE` rule.

The `THRESHOLD` values define the percentage of available space which will trigger a `lowDiskSpace` event, and the percentage of available space at which point the `lowDiskSpace` event will no longer be triggered. There is an additional event, `noDiskSpace`, which is triggered when a storage pool is out of space but this event occurs without the need to define it in a policy rule. The `MIGRATE` rule, and the associated `THRESHOLD` values can be used to implement ILM policies.

The following actions are required to implement the migration policies:

1. Define an appropriate policy using the `MIGRATE` rule and `THRESHOLD` parameters.

2. Install the defined policy into the file system policy rules using the **mmchpolicy** command. For more information, see the topic *mmchpolicy command* in the *IBM Storage Scale: Command and Programming Reference Guide*.
3. Create a callback script that will be invoked for the lowDiskSpace or noDiskSpace events (they can be different scripts) and then register that script using the **mmaddcallback** command. For more information, see the topic *mmaddcallback command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Note: IBM Storage Scale provides the **mmstartpolicy** script which can be used as the callback script for the lowDiskSpace and noDiskSpace events, or customers can choose to implement their own callback script.

A callback must be added to trigger the policy run when the low space event is generated. The **mmstartpolicy** command is provided to start the **mmapplypolicy** run from the administrator-defined callback.

To add a callback, run this command. The following command is on one line:

```
mmaddcallback lowSpaceHandler --event LowDiskSpace --command /usr/lpp/mmf/bin/mmstartpolicy --  
parms "%eventName% %fsName%" --single-instance"
```

The **--single-instance** flag is required to avoid running multiple migrations on the file system at the time.

Related concepts

[Improving performance in very large file systems](#)

Read about how to improve the performance of the **mmapplypolicy** command in very large file systems.

Related tasks

[Creating a policy](#)

Create a text file for your policy by following these guidelines.

[Installing a policy](#)

Install a policy by following these guidelines.

[Changing the active policy](#)

When you prepare a file with the new or changed policy rules, then issue the **mmchpolicy** command.

[Listing policies](#)

When you use the **mmlspolicy** command to list policies, follow these guidelines.

[Validating policies](#)

When you validate a policy file, follow this guideline.

[Deleting policies](#)

When you remove the current policy rules and restore the file-placement policy, follow this guideline.

[Improving performance with the --sort-command parameter](#)

To improve performance of the **mmapplypolicy** command, follow these guidelines.

Improving performance with the --sort-command parameter

To improve performance of the **mmapplypolicy** command, follow these guidelines.

One possible way to improve the performance of the **mmapplypolicy** command is to specify an alternative sort command to be used instead of the default sort command provided by the operating system. To do this, issue **mmapplypolicy --sort-command SortCommand**, specifying the executable path of the alternative command.

For example, on AIX the GNU **sort** program, freely available within the **coreutils** package from [AIX Toolbox for Linux Applications](#) (www.ibm.com/systems/power/software/aix/linux/toolbox), will typically perform large sorting tasks much faster than the standard AIX **sort** command. If you wanted to specify the GNU **sort** program, you would use the following command: **mmapplypolicy --sort-command /opt/freeware/bin/sort**.

Before issuing `mmapplypolicy --sort`-command on a large number of files, first do a performance comparison between the alternative sort command and the default sort command on a smaller number of files to determine whether the alternative command is in fact faster.

If you specify an alternative sort command, it is recommended that you install it on all cluster nodes.

Related concepts

[Using thresholds to migrate data between pools](#)

With the use of storage pools, the potential for a pool to exhaust its available space while other pools have free space can occur.

[Improving performance in very large file systems](#)

Read about how to improve the performance of the `mmapplypolicy` command in very large file systems.

Related tasks

[Creating a policy](#)

Create a text file for your policy by following these guidelines.

[Installing a policy](#)

Install a policy by following these guidelines.

[Changing the active policy](#)

When you prepare a file with the new or changed policy rules, then issue the `mmchpolicy` command.

[Listing policies](#)

When you use the `mmlspolicy` command to list policies, follow these guidelines.

[Validating policies](#)

When you validate a policy file, follow this guideline.

[Deleting policies](#)

When you remove the current policy rules and restore the file-placement policy, follow this guideline.

Improving performance in very large file systems

Read about how to improve the performance of the `mmapplypolicy` command in very large file systems.

The following actions can improve performance:

- Put the **system** pool on the fastest storage available, which can be either solid-state storage or hard disks. In either case, spread the system pool over multiple storage devices, so that they can seek in parallel, independently of one another. Verify that logical disks do not map to the same physical disk.

The policy engine requires many I/O operations against file system metadata. Storing metadata on the fastest storage possible can improve the performance of policy execution.

- Include both the **-N** parameter and the **-g** parameter on the `mmapplypolicy` command line.
 - The **-N** parameter specifies a list of nodes that run parallel instances of policy code.
 - The **-g** parameter specifies a global work directory that can be accessed by the nodes that are specified by the **-N** parameter.
 - When both **-N** and **-g** are specified, `mmapplypolicy` uses high-performance and fault-tolerant protocols during execution.
- If the exact order in which files are processed is not important, consider specifying the **--choice-algorithm fast** algorithm, which works with the **-N** and **-g** options for parallel processing.
- If the order in which files are processed is not important at all, specify **WEIGHT(0)** in your **MIGRATE**, **LIST**, and **DELETE** policy rules.
- Update the file system format to format level 13.01 (GPFS 3.5.0.1) or higher. File systems at this level can support the following two features, among others:
 - Storing small directories and small files in the inode.
 - Fast extended attributes. For this feature, you must also update the file system by running `mmigratefs`.

These two features can improve the performance of the **mmapplypolicy** command. See the following links:

[Chapter 27, “File system format changes between versions of IBM Storage Scale,” on page 267](#)

Completing the migration to a new level of IBM Storage Scale in the *IBM Storage Scale: Administration Guide*

Related concepts

[Using thresholds to migrate data between pools](#)

With the use of storage pools, the potential for a pool to exhaust its available space while other pools have free space can occur.

Related tasks

[Creating a policy](#)

Create a text file for your policy by following these guidelines.

[Installing a policy](#)

Install a policy by following these guidelines.

[Changing the active policy](#)

When you prepare a file with the new or changed policy rules, then issue the **mmchpolicy** command.

[Listing policies](#)

When you use the **mmlspolicy** command to list policies, follow these guidelines.

[Validating policies](#)

When you validate a policy file, follow this guideline.

[Deleting policies](#)

When you remove the current policy rules and restore the file-placement policy, follow this guideline.

[Improving performance with the --sort-command parameter](#)

To improve performance of the **mmapplypolicy** command, follow these guidelines.

Working with external storage pools

With external storage pools you can migrate files to storage pools managed by an external application such as IBM Storage Protect.

The following topics describe how to work with external storage pools:

- [Defining the external pools](#)
- [“User-provided program for managing external pools” on page 572](#)
- [“File list format” on page 572](#)
- [“Record format ” on page 573](#)
- [“Migrate and recall with external pools” on page 574](#)
- [“Pre-migrating files with external storage pools” on page 575](#)
- [“Purging files from external storage pools” on page 575](#)
- [“Using thresholds to migrate data between pools” on page 567](#)

Related concepts

[Overview of policies](#)

A *policy* is a set of rules that describes the life cycle of user data based on the attributes of files. Each rule defines an operation or definition, such as “migrate to a pool and replicate the file.”

[Policy rules](#)

A *policy rule* is an SQL-like statement that tells GPFS what to do with the data for a file in a specific storage pool if the file meets specific criteria. A rule can apply to any file being created or only to files being created within a specific fileset or group of filesets.

[The **mmapplypolicy** command and policy rules](#)

The `mmapplypolicy` command has policy rules that are based on the characteristics of different phases.

Backup and restore with storage pools

When you back up data or restore data to a storage pool, consider the following descriptions.

ILM for snapshots

ILM for snapshots can be used to migrate snapshot data.

Related tasks

Managing policies

Policies and the rules that they contain are used to assign files to specific storage pools.

Related reference

Policy rules: Examples and tips

Before you write and apply policies, consider the following advice.

Defining external pools

When you define external pools, follow these rules.

GPFS file management policy rules control data migration into external storage pools. Before you write a migration policy, you must define the external storage pool that the policy references. After you define the storage pool, you can then create policies that set thresholds that trigger data migration into or out of the referenced external pool.

When a storage pool reaches the defined threshold or when you invoke `mmapplypolicy`, GPFS processes the metadata, generates a list of files, and invokes a user provided script or program which initiates the appropriate commands for the external data management application to process the files. This allows GPFS to transparently control offline storage and provide a tiered storage solution that includes tape or other media.

Before you migrate data to an external storage pool, you must define that pool. To define external storage pools, use a GPFS policy rule as follows:

```
RULE EXTERNAL POOL 'PoolName' EXEC 'InterfaceScript' [OPTS 'OptionsString'] [ESCAPE  
'SpecialCharacters']
```

Where:

- *PoolName* defines the name of the storage pool
- *InterfaceScript* defines the program or script to be invoked to migrate data to or from the external pool
- *OptionsString* is an optional string that, if provided, will be passed to the *InterfaceScript*

You must have a separate EXTERNAL POOL rule for each external pool that you wish to define.

Example of a rule that defines a storage pool

The following rule defines a storage pool called `externalpoolA`.

```
RULE EXTERNAL POOL 'externalpoolA' EXEC '/usr/hsm/bin/hsmControl' OPTS '-server=hsm-manager.nyc.com'
```

In this example:

- `externalpoolA` is the name of the external pool
- `/usr/hsm/bin/hsmControl` is the location of the executable script that will be invoked when there are files for migration
- `-server=hsm-manager.nyc.com` is the location of storage pool `externalpoolA`

For more information, refer to [“User-provided program for managing external pools” on page 572](#).

User-provided program for managing external pools

After you define an external storage pool, subsequent migration or deletion rules might refer to that pool as a source or target storage pool.

When the `mmapplypolicy` command is invoked and a rule dictates that data should be moved to or from an external pool, the user provided program that is identified with the **EXEC** clause in the policy rule starts. That executable program receives three arguments:

- The command to be executed. Your script should implement each of the following sub-commands:
 - LIST - Provides arbitrary lists of files with no semantics on the operation.
 - MIGRATE - Migrate files to external storage and reclaim the online space allocated to the file.
 - PREMIGRATE - Migrate files to external storage but do not reclaim the online space.
 - PURGE - Delete files from both the online file system and the external storage.
 - RECALL - Recall files from external storage to the online storage.
 - TEST – Test for presence and operation readiness. Return zero for success. Return non-zero if the script should not be used on a given node.
- The name of a file containing a list of files to be migrated, premigrated, or purged. See “[File list format](#)” on page 572 for detailed description of the layout of the file.
- Any optional parameters specified with the OPTS clause in the rule. These optional parameters are not interpreted by the GPFS policy engine.

The `mmapplypolicy` command invokes the external pool script on all nodes in the cluster that have installed the script in its designated location. The script must be installed at the node that runs `mmapplypolicy`. You can also install the script at other nodes for parallel operation but that is not required. GPFS may call your exit script one or more times for each command.

Important: Use the EXCLUDE rule to exclude any special files that are created by an external application. For example, when using IBM Storage Protect, exclude the **.SpaceMan** directory to avoid migration of **.SpaceMan**, which is an IBM Storage Protect repository.

Note: The script executed by the **EXEC** clause will inherit the library search path as defined by the policy engine. This search path includes directories under the IBM Storage Scale installation directory to ensure correct library dependencies are used for IBM Storage Scale commands. The library search path can cause unexpected errors if the script invoked by the **EXEC** clause uses commands provided by the IBM Storage Protect product.

If your script uses IBM Storage Protect commands you are advised to either unset the library search path (`LD_LIBRARY_PATH` on Linux and `LIBPATH` on AIX), or ensure it is defined as required by the IBM Storage Protect product.

For example, if InterfaceScript invokes `/usr/bin/dsimmigrate`
on Linux: `LD_LIBRARY_PATH= /usr/bin/dsimmigrate ...`
on AIX: `LIBPATH= /usr/bin/dsimmigrate ...`

File list format

Each call to the external pool script specifies the pathname for a temporary file that contains a list of files to be operated on.

This file list defines one file per line as follows:

```
InodeNumber GenNumber SnapId [OptionalShowArgs] -- FullPathToFile
```

where:

- *InodeNumber* is a 64-bit inode number.
- *GenNumber* is a 32-bit file generation number.
- *SnapId* is a 64-bit snapshot identifier.

- *OptionalShowArgs* is the result, if any, from the evaluation of the SHOW clause in the policy rule.
- *FullPathToFile* is a fully qualified path name to the file. When there are multiple paths within a file system to a particular file (*Inode*, *GenNumber*, and *SnapId*), each path is shown.
- The "--" characters are a field delimiter that separates the optional show parameters from the path name to the file.

Note: GPFS does not restrict the character set used for path and file names. All characters except '\0' are valid. To make the files readily parseable, files or directories containing the newline character and/or other special characters are "escaped", as described previously, in connection with the ESCAPE '%special-characters' clause.

Record format

The format of the records in each file list file can be expressed as shown in the following example.

Each file list file:

```
iAggregate:WEIGHT:INODE:GENERATION:SIZE:iRule:resourceID:attr_flags:  
path-length!PATH_NAME:pool-length!POOL_NAME  
[;show-length!SHOW]end-of-record-character
```

where:

- *iAggregate* is a grouping index that is assigned by `mmapplypolicy`.
- *WEIGHT* represents the WEIGHT policy language file attribute.
- *INODE* represents the INODE policy language file attribute.
- *GENERATION* represents the GENERATION policy language file attribute.
- *SIZE* represents the SIZE policy language file attribute.
- *iRule* is a rule index number assigned by `mmapplypolicy`, which relates to the policy rules file that is supplied with the -P argument.
- *resourceID* represents a pool index, USER_ID, GROUP_ID, or fileset identifier, depending on whether thresholding is done with respect to pool usage or to user, group, or fileset quotas.
- *attr_flags* represents a hexadecimal encoding of some of the attributes that are also encoded by the policy language variable *MISC_ATTRIBUTES*. The low-order 20 bits of *attr_flags* are taken from the *ia_flags* word that is defined in the *gpfs.h* API definition.
- *path-length* represents the length of the character string PATH_NAME.
- *pool-length* represents the length of the character string POOL_NAME.
- *show-length* represents the length of the character string SHOW.
- *end-of-record-character* is \n or \0.

Note: You can only change the values of the *iAggregate*, *WEIGHT*, *SIZE*, and *attr_flags* fields. Changing the values of other fields can cause unpredictable policy execution results.

All of the numeric fields are represented as hexadecimal strings, except the *path-length*, *pool-length*, and *show-length* fields, which are decimal encoded. These fields can be preceded by a minus sign (-), which indicates that the string that follows it contains escape sequences. In this case, the string might contain occurrences of the character pair \n, which represents a single newline character with a hexadecimal value of 0xA in the filename. Also, the string might contain occurrences of the character pair \\, which represents a single \ character in the filename. A \ will only be represented by \\ if there are also newline characters in the filename. The value of the length field within the record counts any escape characters.

The encoding of *WEIGHT* is based on the 64-bit IEEE floating format, but its bits are *flipped* so that when a file list is sorted using a conventional collating sequence, the files appear in decreasing order, according to their *WEIGHT*.

The encoding of *WEIGHT* can be expressed and printed using C++ as:

```
double w = - WEIGHT;
/* This code works correctly on big-endian and little-endian systems */
uint64 u = *(uint64*)&w; /* u is a 64 bit long unsigned integer
                           containing the IEEE 64 bit encoding of the double floating point
                           value of variable w */
uint64 hibit64 = ((uint64)1<<63);
if (w < 0.0) u = ~u; /* flip all bits */
else u = u | hibit64; /* force the high bit from 0 to 1,
                        also handles both "negatively" and "positively" signed 0.0 */
printf("%016llx",u);
```

The format of the majority of each record can be expressed in C++ as:

```
printf("%03x:%016llx:%016llx:%llx:%lx:%x:%11x:%d!%s:%d!%s",
      iAggregate, u /*encoding of -1*WEIGHT from above*/, INODE, ... );
```

Notice that the first three fields are fixed in length to facilitate the sorting of the records by the field values *iAggregate*, *WEIGHT*, and *INODE*.

The format of the optional SHOW string portion of the record can be expressed as:

```
if(SHOW && SHOW[0]) printf(";%d!%s",strlen(SHOW),SHOW);
```

For more information, see the topic *mmapplypolicy command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Migrate and recall with external pools

After you define an external storage pool, subsequent migration or deletion rules might refer to that pool as a source or target storage pool.

When you invoke *mmapplypolicy* and a rule dictates that data should be deleted or moved to or from an external pool, the program identified in the EXTERNAL POOL rule is invoked with the following arguments:

- The command to be executed.
- The name of the file containing a list of files to be migrated, premigrated, or purged.
- Optional parameters, if any.

For example, let us assume an external pool definition:

```
RULE EXTERNAL_POOL 'externalpoolA'
  EXEC '/usr/hsm/bin/hsmControl' OPTS '-server=hsm-manager.nyc.com'
```

Note: The policy SQL parser normally performs M4 macro preprocessing with square brackets set as the single quotation marks. Therefore, it is recommended that you add an extra set of square brackets if square brackets are used in OPTS.

To move files from the internal system pool to storage pool *externalpoolA* you would simply define a migration rule that may look something like this:

```
RULE 'MigToExt' MIGRATE FROM POOL('system') TO POOL('externalpoolA') WHERE ...
```

This would result in the external pool script being invoked as follows:

```
/usr/hsm/bin/hsmControl MIGRATE /tmp/filelist -server=hsm-manager.nyc.com
```

Similarly, a rule to migrate data from an external pool back to an internal storage pool could look like:

```
RULE 'MigFromExt' MIGRATE FROM POOL 'externalpoolA' TO POOL 'system' WHERE ...
```

This would result in the external pool script being invoked as follows:

```
/usr/hsm/bin/hsmControl RECALL /tmp/filelist -server=hsm-manager.nyc.com
```

Notes:

1. When migrating to an external storage pool, GPFS ignores the LIMIT and REPLICATION clauses in the policy rule.
2. If you are using IBM Storage Protect with external storage pools, you may need to create specific rules to avoid system problems. These rules should exclude IBM Storage Protect-related system files from both migration and deletion. These rules use the form:

```
RULE 'exclude hsm system files' EXCLUDE WHERE PATH_NAME LIKE '%/.SpaceMan%'
```

Pre-migrating files with external storage pools

Pre-migration is a standard technique of Hierarchical Storage Management (HSM) systems such as IBM Storage Protect.

Pre-migration copies data from GPFS internal storage pools to external pools but leaves the original data online in the active file system. Pre-migrated files are often referred to as "dual resident" to indicate that the data for the files are available both online in GPFS and offline in the external storage manager. Files in the pre-migrated state allow the external storage manager to respond more quickly to low space conditions by simply deleting the copy of the file data that is stored online.

The files to be pre-migrated are determined by the policy rules that migrate data to an external storage pool. The rule will select files to be migrated and optionally select additional files to be pre-migrated. The THRESHOLD clause of the rule determines the files that need to be pre-migrated.

If you specify the THRESHOLD clause in file migration rules, the mmapplypolicy command selects files for migration when the affected storage pool reaches the specified high occupancy percentage threshold. Files are migrated until the storage pool utilization is reduced to the specified low occupancy percentage threshold. When migrating to an external storage pool, GPFS allows you to specify a third pool occupancy percentage which defines the file pre-migration threshold: after the low occupancy percentage is reached, files are pre-migrated until the pre-migration occupancy percentage is reached.

To explain thresholds in another way, think of an internal storage pool with a high threshold of 90%, a low threshold of 80%, and a pre-migrate threshold of 60%. When this internal storage pool reaches 90% occupancy, the policy rule will migrate files until the occupancy of the pool reaches 80% then it will continue to pre-migrate another 20% of the file space until the 60% threshold is reached.

Pre-migration can only be done with external storage managers using the XDSM Data Storage Management API (DMAPI). Files in the migrated and pre-migrated state will have a DAPI managed region set on the file data. Files with a managed region are visible to mmapplypolicy and may be referenced by a policy rule. You can approximate the amount of pre-migrated space required by counting the space used after the end of the first full data block on all files with managed regions.

Note:

1. If you do not set a pre-migrate threshold or if you set a value that is greater than or equal to the low threshold, then GPFS will not pre-migrate files. This is the default setting.
2. If you set the pre-migrate threshold to zero, then GPFS will pre-migrate all files.

Purging files from external storage pools

Files that have been migrated to an external storage pool continue to have their file name and attributes stored in GPFS; only the file data has been migrated. Files that have been migrated or pre-migrated to an external storage pool may be deleted from the GPFS internal storage pool and from the external storage pool with the policy language using a DELETE rule.

```
RULE 'DelFromExt' DELETE WHERE ...
```

If the file has been migrated or pre-migrated, this would result in the external pool script being invoked as follows:

```
/usr/hsm/bin/hsmControl PURGE /tmp/filelist -server=hsm-manager.nyc.com
```

The script should delete a file from both the online file system and the external storage manager. However, most IBM Storage Protect systems automatically delete a file from the external storage manager whenever the online file is deleted. If that is how your IBM Storage Protect system functions, your script will only have to delete the online file.

Backup and restore with storage pools

When you back up data or restore data to a storage pool, consider the following descriptions.

You can use the GPFS ILM tools to back up data for disaster recovery or data archival to an external storage manager such as the IBM Storage Protect Backup-Archive client. When backing up data, the external storage manager must preserve the file name, attributes, extended attributes, and the file data. Among other things, the extended attributes of the file also contain information about the assigned storage pool for the file. When you restore the file, this information is used to assign the storage pool for the file data.

The file data might be restored to the storage pool to which it was assigned when it was backed up. Otherwise, it might be restored to a pool selected by a restore or placement rule by using the backed-up attributes for the file. GPFS supplies three subroutines that support backup and restore functions with external pools:

- `gpfs_fgetattr()`
- `gpfs_fputattr()`
- `gpfs_fputattrswithpathname()`

GPFS exports the extended attributes for a file, including its ACLs, using `gpfs_fgetattr()`. Included in the extended attributes is the name of the storage pool to which the file is assigned, and file attributes that are used for file placement. The file is restored the extended attributes are restored using either `gpfs_fputattr()` or `gpfs_fputattrswithpathname()`.

When a backup application uses `gpfs_fputattr()` to restore the file, GPFS assigns the restored file to the storage pool with the same name as when the file was backed up. Thus, by default, restored files are assigned to the same storage pool they were in when they were backed up. If that pool is not available, GPFS tries to select a pool using the current file placement rules. If that fails, GPFS assigns the file to the system storage pool.

Note: If a backup application uses `gpfs_fputattr()` to restore a file, it omits the RESTORE RULE.

When a backup application restores the file using `gpfs_fputattrswithpathname()`, the GPFS is able to access more file attributes that are used by placement or migration policy rules to select the storage pool for the file. This information includes the UID and GID for the owner, the access time for the file, file modification time, file size, the amount of storage allocated, and the full path to the file. GPFS uses `gpfs_fputattrswithpathname()` to match this information with restore policy rules that you define.

In other words, the RESTORE rule looks at saved file attributes rather than the current file attributes. The call to `gpfs_fputattrswithpathname()` tries to match the saved information to a RESTORE rule. If the RESTORE rules cannot match saved attributes, GPFS tries to restore the file to the same storage pool it was in when the file was backed up. If that pool is not available GPFS tries to select a pool by matching placement rules. If that fails, GPFS assigns the file to the system storage pool.

Note: When a RESTORE rule is used, and restoring the file to the specified pool would exceed the occupancy percentage that is defined for that pool, GPFS skips that rule and the policy engine looks for the next rule that matches. While testing for matching rules, GPFS considers the specified replication factor and the KB_ALLOCATED attribute of the file that is being restored.

The `gpfs_fgetattr()`, `gpfs_fputattr()`, and `gpfs_fputattrswithpathname()` subroutines have optional flags that further control the selection of storage pools. For more information, see the topics `gpfs_fgetattr()` subroutine, `gpfs_fputattr()` subroutine, and `gpfs_fputattrswithpathname()` subroutine in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

[Overview of policies](#)

A *policy* is a set of rules that describes the life cycle of user data based on the attributes of files. Each rule defines an operation or definition, such as "migrate to a pool and replicate the file."

Policy rules

A *policy rule* is an SQL-like statement that tells GPFS what to do with the data for a file in a specific storage pool if the file meets specific criteria. A rule can apply to any file being created or only to files being created within a specific fileset or group of filesets.

The mmapplypolicy command and policy rules

The `mmapplypolicy` command has policy rules that are based on the characteristics of different phases.

Working with external storage pools

With external storage pools you can migrate files to storage pools managed by an external application such as IBM Storage Protect.

ILM for snapshots

ILM for snapshots can be used to migrate snapshot data.

Related tasks

Managing policies

Policies and the rules that they contain are used to assign files to specific storage pools.

Related reference

Policy rules: Examples and tips

Before you write and apply policies, consider the following advice.

Working with external lists

External lists, like external pools, generate lists of files. For external pools, the operations on the files correspond to the rule that references the external pool. For external lists, there is no implied operation; it is simply a list of files that match the criteria specified in the policy rule.

External lists must be defined before they can be used. External lists are defined by:

```
RULE EXTERNAL LIST 'ListName' EXEC 'InterfaceScript' [OPTS 'OptionsString'] [ESCAPE 'SpecialCharacters']
```

Where:

- *ListName* defines the name of the external list
- *InterfaceScript* defines the program to be invoked to operate on the list of files
- *OptionsString* is an optional string that, if provided, will be passed to the *InterfaceScript*

See ["User-provided program for managing external pools" on page 572](#).

Example

The following rule defines an external list called `listfiles`:

```
RULE EXTERNAL LIST 'listfiles' EXEC '/var/mmfs/etc/listControl' OPTS '-verbose'
```

In this example:

- `listfiles` is the name of the external list
- `/var/mmfs/etc/listControl` is the location of the executable script that defines the operations on the list of files
- `-verbose` is an optional flag to the `listControl` script

The EXTERNAL LIST rule provides the binding between the lists generated with regular LIST rules and the external program that you want to run with these lists as input. For example, this rule would generate a list of all files that have more than 1 MB of data in an internal storage pool:

```
RULE 'ListLargeFiles' LIST 'listfiles' WHERE KB_ALLOCATED > 1024
```

By default, only user files are included in lists. To include directories, symbolic links, and other file system objects, the DIRECTORIES_PLUS clause must be specified. For example, this rule would generate a list of all objects in the file system.

```
RULE 'ListAllObjects' LIST 'listfiles' DIRECTORIES_PLUS
```

ILM for snapshots

ILM for snapshots can be used to migrate snapshot data.

Similar to the files in the root file system, snapshot data can also be managed by using policy rules. Rules can be written to migrate snapshot data among internal storage pools or generated in specific pools.

Snapshot data migration

Snapshot data can be migrated by using the **mmapplypolicy** command with simple migration rules. For example, to migrate data of a snapshot with the name snapname from an SSD pool to the Capacity pool, use the following rule:

```
RULE 'MigToCap' MIGRATE FROM POOL 'SSD' TO POOL 'Capacity'
```

Then, run the **mmapplypolicy** command with the **-S snapname** parameter to complete the migration.

Snapshot data belonging to AFM and AFM DR can also be migrated. Use the following rule:

```
RULE 'migrate' MIGRATE FROM POOL 'POOL1' TO POOL 'POOL2'
```

In this example, data is migrated from POOL1 to POOL2. You must exclude files which are internal to AFM while migrating snapshot data. An example of a rule to exclude such files is as under:

```
RULE 'migrate' MIGRATE FROM POOL 'POOL1' TO POOL 'POOL2' WHERE  
( NOT (PATH_NAME LIKE '/%/.afm%') OR (PATH_NAME LIKE '/%/.ptrash%')  
OR (PATH_NAME LIKE '/%/.afmtrash%')OR (PATH_NAME LIKE '/%/.pconflicts%'))
```

Note:

- The snapshot data cannot be migrated to external pools.
- The migration rules for snapshot data cannot be mixed with other rule types.
- SetXattr file function is not allowed on both the MIGRATE and SET_SNAP_POOL rules for snapshot files.

Snapshot data placement

A snapshot placement rule can be used to generate snapshot data in specific internal pools. For example, to generate the snapshot data for all snapshots in the Capacity pool, use the following rule:

```
RULE 'SnapPlacement' SET SNAP_POOL 'Capacity'
```

Snapshot data for specific snapshots can be placed in specific pools by using the following rule:

```
RULE 'SnapPlacement' SET SNAP_POOL 'Capacity' WHERE SNAP_NAME LIKE '%daily%'
```

Include this rule in the set of rules installed for the file system. Placement of a snapshot file happens when the first data block is copied to it because of the changes made to the file in the root file system.

Note: Snapshot data cannot be placed in external pools.

The placement rule can be applied to snapshot data belonging to AFM and AFM DR. In the following example, snap pool is set as POOL1, for all snapshots having psnap as a sub-pattern in the name.

```
RULE 'setsnappool' SET SNAP_POOL 'POOL1' WHERE SNAP_NAME LIKE '%psnap%'
```

Another example is as under -

```
RULE 'setsnappool' SET SNAP_POOL 'POOL1' WHERE SNAP_NAME LIKE '%afm%'
```

Ill placement of snapshot files

Deletion of files can result in these files moving to snapshot and then becoming ill-placed or ill-replicated. In these cases, the **mmrestripefile** command can be used to correct the ill placement and ill replication of snapshot files.

After the **SNAP_POOL** variable is defined in the placement policy, the data blocks for files that are deleted from the file system remain there as ill-placed. These files are not copied to the storage pool defined by the **SNAP_POOL** variable. Additionally, users must run the **mmrestripefs** command (or possibly the **mmrestripefs**) command to move the ill-placed data blocks to the pool defined by the **SNAP_POOL** variable.

Related concepts

Overview of policies

A *policy* is a set of rules that describes the life cycle of user data based on the attributes of files. Each rule defines an operation or definition, such as "migrate to a pool and replicate the file."

Policy rules

A *policy rule* is an SQL-like statement that tells GPFS what to do with the data for a file in a specific storage pool if the file meets specific criteria. A rule can apply to any file being created or only to files being created within a specific fileset or group of filesets.

The mmapplypolicy command and policy rules

The **mmapplypolicy** command has policy rules that are based on the characteristics of different phases.

Working with external storage pools

With external storage pools you can migrate files to storage pools managed by an external application such as IBM Storage Protect.

Backup and restore with storage pools

When you back up data or restore data to a storage pool, consider the following descriptions.

Related tasks

Managing policies

Policies and the rules that they contain are used to assign files to specific storage pools.

Related reference

Policy rules: Examples and tips

Before you write and apply policies, consider the following advice.

User storage pools

All user data for a file is stored in the assigned storage pool as determined by your file placement rules.

In addition, file data can be migrated to a different storage pool according to your file management policies. For more information on policies, see ["Policies for automating file management" on page 531](#).

A user storage pool contains only the blocks of data (user data, for example) that make up a user file. GPFS stores the data that describes the files, called *file metadata*, separately from the actual file data in the system storage pool. You can create one or more user storage pools, and then create policy rules to indicate where the data blocks for a file should be stored.

For more information see ["File heat: Tracking file access temperature" on page 579](#).

File heat: Tracking file access temperature

A file's access temperature is an attribute for policy that provides a means of optimizing tiered storage.

File temperatures are a relative attribute, indicating whether a file is "hotter" or "colder" than the others in its pool. The policy can be used to migrate hotter files to faster tiers and colder files to slower tiers.

The access temperature is an exponential moving average of the accesses to the file. As files are accessed

the temperature increases; likewise when the access stops the file cools. File temperature is intended to optimize nonvolatile storage, not memory usage; therefore, cache hits are not counted. In a similar manner, only user accesses are counted.

The access counts to a file are tracked as an exponential moving average. An unaccessed file loses a percentage of its accesses each period. The loss percentage and tracking period are set by the following configuration values:

fileHeatPeriodMinutes

A nonzero value enables file access temperature tracking and specifies the frequency with which the file heat attributes are updated. A value of 0 disables FILE_HEAT tracking. The default value is 0.

fileHeatLossPercent

This attribute specifies the percent of file access heat that an unaccessed file loses at the end of each tracking period. The valid range is 0 - 100. The default value is 10. The tracking period is set by **fileHeatPeriodMinutes**.

For more information, see the topic *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

File Heat can be configured on a per-cluster basis, not on a per-file system basis. Use WEIGHT(FILE_HEAT) with a policy MIGRATE rule to prioritize migration by file temperature. (You can use the GROUP POOL rule to define a group pool to be specified as the TO POOL target.) See “[Policies for automating file management](#)” on page 531.

Be aware of the following factors:

- New values for **fileHeatPeriodMinutes** and **fileHeatLossPercent** are not effective until the GPFS daemon is stopped and restarted.
- The file heat attribute of a file is accessible externally through the **FILE_HEAT** policy attribute. This attribute is not updated until the update inode of the file is written to the storage media. You can trigger the update by unmounting the file system.
- After a file access, the access temperature of the file is increased when the file access time (**atime**) is set. If the updating of **atime** is suppressed or if relative **atime** semantics are in effect, proper calculation of the file access temperature may be adversely affected.

Object heatmap data tiering policies

For more information see the following topics:

Object heatmap data tiering in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
[“Enabling the object heatmap policy” on page 422](#)

Examples

1. The following command sets **fileHeatPeriodMinutes** to 1440 (24 hours) and **fileHeatLossPercent** to 10, meaning that unaccessed files lose 10% of their heat value every 24 hours, or approximately 0.4% every hour (because the loss is continuous and increases geometrically):

```
mmchconfig fileheatperiodminutes=1440,fileheatlosspercent=10
```

2. You can test file heat with the following steps:

- a. Issue the **mmchconfig** command to set the file heat tracking period to 60 minutes. This action enables file access temperature tracking for the cluster if it is not already enabled. You can issue the **mmlsconfig** command to verify that the tracking period is set:

```
#mmchconfig fileHeatPeriodMinutes=60
mmchconfig: Command successfully completed
#mmlsconfig | grep -i heat
fileHeatPeriodMinutes 60
```

- b. Restart the GPFS daemon to make the new **fileHeatPeriodMinutes** value effective.

```
#mmshutdown  
#mmstartup
```

- c. Mount a file system and verify that a file exists:

```
#mmmount c23  
#ls -l /c23/10g  
-rw-r--r--. 1 root root 10737418240 May 16 15:09 /c23/10g
```

- d. Display the file access temperature (file heat) of the file:

```
#mmlsattr -d -X /c23/10g  
file name:          /c23/10g  
security.selinux
```

In this example the file access temperature of the file has never been set, so the **mmlsattr** command does not report any file heat.

- e. Run the **dd** command-line utility to read the file from the storage device:

```
dd if=/c23/10g bs=1M of=/dev/null
```

- f. Wait a short while for the inode of the file to be written to the storage device. You can force this action by unmounting the file system.

- g. Issue the **mmlsattr** command to verify that the file heat is updated:

```
#mmlsattr -d -X /c23/10g  
file name:          /c23/10g  
....  
security.selinux  
gpfs.FileHeat
```

- h. Issue the **mmlsattr** command again to display the hexadecimal values of the cluster attributes:

```
#mmlsattr -d -X -L /c23/10g  
file name:          /c23/10g  
....  
security.selinux:  
....  
gpfs.FileHeat:      0x000000EE42A40400
```

In this example the file heat of the file is 0x000000EE42A40400. The **FILE_HEAT** policy attribute returns this value.

3. Below is an example of migration rules to rebalance data between the pools, moving the hottest data to the platinum pool until it is 70% full, then moving the next hottest data to the gold pool until it is 80% full, then the silver and finally all remaining data to the offline bronze pool.

```
/* Define an external pool for the off-line storage */  
RULE EXTERNAL POOL 'bronze' EXEC ''  
  
/* Define pool group using three on-line pools and the external off-line pool. */  
RULE 'DefineTiers' GROUP POOL 'TIERS'  
      IS 'platinum' LIMIT(70)  
      THEN 'gold' LIMIT(80)  
      THEN 'silver' LIMIT(90)  
      THEN 'bronze'  
  
RULE 'Rebalance' MIGRATE FROM POOL 'TIERS' TO POOL 'TIERS' WEIGHT(FILE_HEAT)  
RULE 'Ingest' MIGRATE FROM POOL 'system' TO POOL 'TIERS' WEIGHT(FILE_HEAT)
```

4. See the **mmapplypolicy-fileheat.sample** script in **/usr/lpp/mmfs/samples/ilm/README**.

Filesets

In most file systems, a file hierarchy is represented as a series of directories that form a tree-like structure. Each directory contains other directories, files, or other file system objects such as symbolic links and hard links. Every file system object has a name associated with it, and is represented in the namespace as a node of the tree.

In addition, GPFS utilizes a file system object called a *fileset*. A fileset is a subtree of a file system namespace that in many respects behaves like an independent file system. Filesets provide a means of partitioning the file system to allow administrative operations at a finer granularity than the entire file system:

- Filesets can be used to define quotas on both data blocks and inodes.
- The owning fileset is an attribute of each file and can be specified in a policy to control initial data placement, migration, and replication of the file's data. See “[Policies for automating file management](#)” on page 531.
- Fileset snapshots can be created instead of creating a snapshot of an entire file system.

GPFS supports independent and dependent filesets. An independent fileset is a fileset with its own inode space. An inode space is a collection of inode number ranges reserved for an independent fileset. An inode space enables more efficient per-fileset functions, such as fileset snapshots. A dependent fileset shares the inode space of an existing, independent fileset. Files created in a dependent fileset are assigned inodes in the same collection of inode number ranges that were reserved for the independent fileset from which it was created.

When the file system is created, only one fileset, called the *root* fileset, exists. The root fileset is an independent fileset that cannot be deleted. It contains the root directory as well as any system files such as quota files. As new files and directories are created, they automatically become part of the parent directory's fileset. The fileset to which a file belongs is largely transparent for ordinary file access, but the containing fileset can be displayed along with the other attributes of each file using the `mmlsattr -L` command.

The root directory of a GPFS file system is also the root of the root fileset.

Fileset namespace

A newly created fileset consists of an empty directory for the root of the fileset, and it is initially not linked into the file system's namespace. A newly created fileset is not visible to the user until it is attached to the namespace by issuing the `mmlinkfileset` command.

Filesets are attached to the namespace with a special link called a *junction*. A junction is a special directory entry, much like a POSIX hard link, that connects a name in a directory of one fileset (source) to the root directory of another fileset (target). A fileset may be the target of only one junction, so that a fileset has a unique position in the namespace and a unique path to any of its directories. The target of the junction is referred to as the *child fileset*, and a fileset can have any number of children. From the user's viewpoint, a junction always appears as if it were a directory, but the user is not allowed to issue the `unlink` or `rmdir` commands on a junction.

Once a fileset has been created and linked into the namespace, an administrator can unlink the fileset from the namespace by issuing the `mmunlinkfileset` command. This makes all files and directories within the fileset inaccessible. If other filesets were linked below it, the other filesets become inaccessible, but they do remain linked and will become accessible again when the fileset is re-linked. Unlinking a fileset, like unmounting a file system, fails if there are open files. The `mmunlinkfileset` command has a force option to close the files and force the unlink. If there are open files in a fileset and the fileset is unlinked with the force option, future references to those files will result in ESTALE errors. Once a fileset is unlinked, it can be re-linked into the namespace at its original location or any other location (it cannot be linked into its children since they are not part of the namespace while the parent fileset is unlinked).

The namespace inside a fileset is restricted to a single, connected subtree. In other words, a fileset has only one root directory and no other entry points such as hard links from directories in other

filesets. Filesets are always connected at the root directory and only the junction makes this connection. Consequently, hard links cannot cross fileset boundaries. Symbolic links, of course, can be used to provide shortcuts to any file system object in the namespace.

The root fileset is an exception. The root fileset is attached to the local namespace using the standard mount command. It cannot be created, linked, unlinked or deleted using the GPFS fileset commands.

For more information about managing filesets, see “[Managing filesets](#)” on page 588.

Related concepts

[Filesets and quotas](#)

The GPFS quota commands support the -j option for fileset block and inode allocation.

[Filesets and storage pools](#)

Filesets are not specifically related to storage pools, although each file in a fileset physically resides in blocks in a storage pool. This relationship is many-to-many; each file in the fileset can be stored in a different user storage pool.

[Filesets and global snapshots](#)

A GPFS global snapshot preserves the contents of the entire file system, including all its filesets, even unlinked ones.

[Fileset-level snapshots](#)

Instead of creating a global snapshot of an entire file system, a fileset snapshot can be created to preserve the contents of a single independent fileset plus all dependent filesets that share the same inode space.

[Filesets and backup](#)

The mmbackup command and IBM Storage Protect are unaware of the existence of filesets. When restoring a file system that had been backed up to IBM Storage Protect, the files are restored to their original path names, regardless of the filesets of which they were originally a part.

Related tasks

[Managing filesets](#)

Filesets and quotas

The GPFS quota commands support the -j option for fileset block and inode allocation.

The quota limit on blocks and inodes in a fileset are independent of the limits for specific users or groups of users. See the following commands in the *IBM Storage Scale: Command and Programming Reference Guide*:

- `mmdefedquota`
- `mmdefquotaon`
- `mmdefquotaoff`
- `mmedquota`
- `mmlsquota`
- `mmquotaoff`
- `mmquotaon`
- `mmrepquota`

In addition, see the description of the --perfileset-quota parameter of the following commands:

- `mmchfs`
- `mmcdfs`
- `mmlsfs`

Important: Quota limits are not enforced for root users (by default). To enforce the quota limits for root users in filesets, change the `enforceFilesetQuotaOnRoot` configuration setting value to yes. For more information, see the `mmchconfig` command.

Related concepts

Fileset namespace

A newly created fileset consists of an empty directory for the root of the fileset, and it is initially not linked into the file system's namespace. A newly created fileset is not visible to the user until it is attached to the namespace by issuing the `mmlinkfileset` command.

Filesets and storage pools

Filesets are not specifically related to storage pools, although each file in a fileset physically resides in blocks in a storage pool. This relationship is many-to-many; each file in the fileset can be stored in a different user storage pool.

Filesets and global snapshots

A GPFS global snapshot preserves the contents of the entire file system, including all its filesets, even unlinked ones.

Fileset-level snapshots

Instead of creating a global snapshot of an entire file system, a fileset snapshot can be created to preserve the contents of a single independent fileset plus all dependent filesets that share the same inode space.

Filesets and backup

The `mmbackup` command and IBM Storage Protect are unaware of the existence of filesets. When restoring a file system that had been backed up to IBM Storage Protect, the files are restored to their original path names, regardless of the filesets of which they were originally a part.

Related tasks

Managing filesets

Filesets and storage pools

Filesets are not specifically related to storage pools, although each file in a fileset physically resides in blocks in a storage pool. This relationship is many-to-many; each file in the fileset can be stored in a different user storage pool.

A storage pool can contain files from many filesets. However, all of the data for a particular file is wholly contained within one storage pool.

Using file-placement policies, you can specify that all files created in a particular fileset are to be stored in a specific storage pool. Using file-management policies, you can define how files in a specific fileset are to be moved or deleted during the file's life cycle. See [“Policy rules: Terms” on page 535](#).

Related concepts

Fileset namespace

A newly created fileset consists of an empty directory for the root of the fileset, and it is initially not linked into the file system's namespace. A newly created fileset is not visible to the user until it is attached to the namespace by issuing the `mmlinkfileset` command.

Filesets and quotas

The GPFS quota commands support the `-j` option for fileset block and inode allocation.

Filesets and global snapshots

A GPFS global snapshot preserves the contents of the entire file system, including all its filesets, even unlinked ones.

Fileset-level snapshots

Instead of creating a global snapshot of an entire file system, a fileset snapshot can be created to preserve the contents of a single independent fileset plus all dependent filesets that share the same inode space.

Filesets and backup

The `mmbackup` command and IBM Storage Protect are unaware of the existence of filesets. When restoring a file system that had been backed up to IBM Storage Protect, the files are restored to their original path names, regardless of the filesets of which they were originally a part.

Related tasks

[Managing filesets](#)

Filesets and global snapshots

A GPFS global snapshot preserves the contents of the entire file system, including all its filesets, even unlinked ones.

The state of filesets in the snapshot is unaffected by changes made to filesets in the active file system, such as unlink, link or delete. The saved file system can be accessed through the `.snapshots` directories and the namespace, including all linked filesets, appears as it did when the snapshot was created. Unlinked filesets are inaccessible in the snapshot, as they were in the active file system. However, restoring a snapshot also restores the unlinked filesets, which can then be re-linked and accessed.

If a fileset is included in a global snapshot, it can be deleted but it is not entirely removed from the file system. In this case, the fileset is emptied of all contents and given a status of 'deleted'. The contents of a fileset remain available in the snapshots that include the fileset (that is, through some path containing a `.snapshots` component) even after the fileset is deleted, since all the contents of the fileset are saved when a snapshot is created. The fileset remains in the deleted state until the last snapshot containing it is deleted, at which time the fileset is automatically deleted.

A fileset is included in a global snapshot if the snapshot is created after the fileset was created. Deleted filesets appear in the output of the `mmlsfileset` and `mmlsfileset --deleted` commands, and the `-L` option can be used to display the latest snapshot that includes a fileset.

During a restore from a global snapshot, attributes of filesets included in the snapshot can be altered. The filesets included in the global snapshot are restored to their former state, and newer filesets are deleted. Also, restore may undelete deleted filesets and change linked filesets to unlinked or vice versa. If the name of a fileset was changed since the snapshot was taken, the old fileset name will be restored.

Related concepts

[Fileset namespace](#)

A newly created fileset consists of an empty directory for the root of the fileset, and it is initially not linked into the file system's namespace. A newly created fileset is not visible to the user until it is attached to the namespace by issuing the `mmlinkfileset` command.

[Filesets and quotas](#)

The GPFS quota commands support the `-j` option for fileset block and inode allocation.

[Filesets and storage pools](#)

Filesets are not specifically related to storage pools, although each file in a fileset physically resides in blocks in a storage pool. This relationship is many-to-many; each file in the fileset can be stored in a different user storage pool.

[Fileset-level snapshots](#)

Instead of creating a global snapshot of an entire file system, a fileset snapshot can be created to preserve the contents of a single independent fileset plus all dependent filesets that share the same inode space.

[Filesets and backup](#)

The `mmbackup` command and IBM Storage Protect are unaware of the existence of filesets. When restoring a file system that had been backed up to IBM Storage Protect, the files are restored to their original path names, regardless of the filesets of which they were originally a part.

Related tasks

[Managing filesets](#)

Fileset-level snapshots

Instead of creating a global snapshot of an entire file system, a fileset snapshot can be created to preserve the contents of a single independent fileset plus all dependent filesets that share the same inode space.

If an independent fileset has dependent filesets that share its inode space, then a snapshot of the independent fileset will also include those dependent filesets. In other words, a fileset snapshot is a snapshot of the whole inode space.

Each independent fileset has its own hidden `.snapshots` directory in the root directory of the fileset that contains any fileset snapshots. The `mmsnapdir` command allows setting an option that makes global snapshots also available through `.snapshots` in the root directory of all independent filesets. The `.snapshots` directory in the file system root directory lists both global snapshots and fileset snapshots of the root fileset (the root fileset is an independent fileset). This behavior can be customized with the `mmsnapdir` command.

Fileset snapshot names need not be unique across different filesets, so it is valid to use the same name for fileset snapshots of two different filesets because they will appear under `.snapshots` in two different fileset root directories.

Note: In order to use the snapshot name in a fileset backup, the name needs to be unique to the file system.

You can restore independent fileset snapshot data and attribute files with the `mmrestorefs` command. For complete usage information, see the topic *mmrestorefs command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

Fileset namespace

A newly created fileset consists of an empty directory for the root of the fileset, and it is initially not linked into the file system's namespace. A newly created fileset is not visible to the user until it is attached to the namespace by issuing the `mmlinkfileset` command.

Filesets and quotas

The GPFS quota commands support the `-j` option for fileset block and inode allocation.

Filesets and storage pools

Filesets are not specifically related to storage pools, although each file in a fileset physically resides in blocks in a storage pool. This relationship is many-to-many; each file in the fileset can be stored in a different user storage pool.

Filesets and global snapshots

A GPFS global snapshot preserves the contents of the entire file system, including all its filesets, even unlinked ones.

Filesets and backup

The `mmbackup` command and IBM Storage Protect are unaware of the existence of filesets. When restoring a file system that had been backed up to IBM Storage Protect, the files are restored to their original path names, regardless of the filesets of which they were originally a part.

Related tasks

Managing filesets

Filesets and backup

The `mmbackup` command and IBM Storage Protect are unaware of the existence of filesets. When restoring a file system that had been backed up to IBM Storage Protect, the files are restored to their original path names, regardless of the filesets of which they were originally a part.

IBM Storage Protect has no mechanism to create or link filesets during restore. Therefore, if a file system is migrated to IBM Storage Protect and then filesets are unlinked or deleted, restore or recall of the file system does not restore the filesets.

During a full restore from backup, all fileset information is lost and all files are restored into the root fileset. It is recommended that you save the output of the `mmlsfileset` command to aid in the reconstruction of fileset names and junction locations. Saving `mmlsfileset -L` also allows reconstruction of fileset comments. Both command outputs are needed to fully restore the fileset configuration.

A partial restore can also lead to confusion if filesets have been deleted, unlinked, or their junctions moved, since the backup was made. For example, if the backed up data was in a fileset that has since been unlinked, the restore process puts it into files and directories in the parent fileset. The unlinked fileset cannot be re-linked into the same location until the restored data is moved out of the way. Similarly, if the fileset was deleted, restoring its contents does not recreate the deleted fileset, but the contents are instead restored into the parent fileset.

Since the `mmbackup` command operates by traversing the directory structure, it does not include the contents of unlinked filesets, even though they are part of the file system. If it is desired to include these filesets in the backup, they should be re-linked, perhaps into a temporary location. Conversely, temporarily unlinking a fileset is a convenient mechanism to exclude it from a backup.

Note: It is recommended not to unlink filesets when doing backups. Unlinking a fileset during an `mmbackup` run can cause the following:

- failure to back up changes in files that belong to an unlinked fileset
- expiration of files that were backed up in a previous `mmbackup` run
- The snapshot name that is used during fileset backup must be unique to the file system.

In summary, fileset information should be saved by periodically recording `mmlsfileset` output somewhere in the file system, where it is preserved as part of the backup process. During restore, care should be exercised when changes in the fileset structure have occurred since the backup was created.



Attention: If you are using the IBM Storage Protect Backup-Archive client you must use caution when you unlink filesets that contain data backed up by IBM Storage Protect. IBM Storage Protect tracks files by pathname and does not track filesets. As a result, when you unlink a fileset, it appears to IBM Storage Protect that you deleted the contents of the fileset. Therefore, the IBM Storage Protect Backup-Archive client inactivates the data on the TSM server which may result in the loss of backup data during the expiration process.

Related concepts

Fileset namespace

A newly created fileset consists of an empty directory for the root of the fileset, and it is initially not linked into the file system's namespace. A newly created fileset is not visible to the user until it is attached to the namespace by issuing the `mmlinkfileset` command.

Filesets and quotas

The GPFS quota commands support the `-j` option for fileset block and inode allocation.

Filesets and storage pools

Filesets are not specifically related to storage pools, although each file in a fileset physically resides in blocks in a storage pool. This relationship is many-to-many; each file in the fileset can be stored in a different user storage pool.

Filesets and global snapshots

A GPFS global snapshot preserves the contents of the entire file system, including all its filesets, even unlinked ones.

Fileset-level snapshots

Instead of creating a global snapshot of an entire file system, a fileset snapshot can be created to preserve the contents of a single independent fileset plus all dependent filesets that share the same inode space.

Related tasks

Managing filesets

Managing filesets

Managing your filesets includes:

- [“Creating a fileset” on page 588](#)
- [“Deleting a fileset” on page 590](#)
- [“Linking a fileset” on page 591](#)
- [“Unlinking a fileset” on page 592](#)
- [“Changing fileset attributes” on page 592](#)
- [“Displaying fileset information” on page 593](#)

Related concepts

Fileset namespace

A newly created fileset consists of an empty directory for the root of the fileset, and it is initially not linked into the file system's namespace. A newly created fileset is not visible to the user until it is attached to the namespace by issuing the `mmlinkfileset` command.

Filesets and quotas

The GPFS quota commands support the `-j` option for fileset block and inode allocation.

Filesets and storage pools

Filesets are not specifically related to storage pools, although each file in a fileset physically resides in blocks in a storage pool. This relationship is many-to-many; each file in the fileset can be stored in a different user storage pool.

Filesets and global snapshots

A GPFS global snapshot preserves the contents of the entire file system, including all its filesets, even unlinked ones.

Fileset-level snapshots

Instead of creating a global snapshot of an entire file system, a fileset snapshot can be created to preserve the contents of a single independent fileset plus all dependent filesets that share the same inode space.

Filesets and backup

The `mmbackup` command and IBM Storage Protect are unaware of the existence of filesets. When restoring a file system that had been backed up to IBM Storage Protect, the files are restored to their original path names, regardless of the filesets of which they were originally a part.

Creating a fileset

Filesets are created with the `mmcrlfileset` command.

By default, filesets are created as dependent filesets that share the inode space of the root. The `--inode-space ExistingFileset` option can be used to create a dependent fileset that shares inode space with an existing fileset. The `--inode-space new` option can be used to create an independent fileset with its own dedicated inode space.

A newly created fileset consists of an empty directory for the root of the fileset and it is initially not linked into the existing namespace. Consequently, a new fileset is not visible and files cannot be added to it, but the fileset name is valid and the administrator can establish quotas on it or policies for it. The administrator must link the fileset into its desired location in the file system's namespace by issuing the `mmlinkfileset` command in order to make use of it.

After the fileset is linked, the administrator can change the ownership and permissions for the new root directory of the fileset, which default to `root` and `0700`, to allow users access to it. Files and directories copied into or created within the directory of the fileset become part of the new fileset.

Note the following restrictions on fileset names:

- The name must be unique within the file system.

- The length of the name must be in the range 1-255.
- The name `root` is reserved for the fileset of the root directory of the file system.
- The name cannot be the reserved word `new`. However, the character string `new` can appear within a fileset name.
- The name cannot begin with a hyphen (-).
- The name cannot contain the following characters: / ? \$ & * () ` # | [] \
- The name cannot contain a white-space character such as blank space or tab.

For more information, see the topics `mmcrfileset command` and `mmlinkfileset command` in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related tasks

Deleting a fileset

Filesets are deleted with the `mmdelfileset` command.

Linking a fileset

After the fileset is created, a junction must be created to link it to the desired location in the file system's namespace using the `mmlinkfileset` command.

Unlinking a fileset

A junction to a fileset is removed with the `mmunlinkfileset` command, which unlinks the fileset only from the active directory namespace. The linked or unlinked state of a fileset in a snapshot is unaffected. The unlink fails if there are files open in the fileset, unless the `-f` option is specified. The root fileset cannot be unlinked.

Changing fileset attributes

To change a junction to a fileset, you have to first unlink the fileset using the `mmunlinkfileset` command, and then create the new junction using the `mmlinkfileset` command.

Displaying fileset information

Fileset status and attributes are displayed with the `mmlsfileset` command.

Creating a fileset by using GUI

You can create a fileset to partition a file system for administrative operations at a finer granularity than the entire file system.

You can create the following two types of filesets:

- Independent filesets
- Dependent filesets

A dependent fileset cannot be changed into an independent fileset, or vice versa. Independent filesets have all the capabilities of a dependent fileset.

Perform the following steps to create a fileset:

1. Go to **Files > Filesets** page in the IBM Storage Scale GUI.
2. Click **Create Fileset**. The **Create Fileset** window appears.
3. Select **Basic** or **Custom** in the Create Fileset window. By using the *Basic* option, you can define only the basic attributes of the fileset. In the *Custom* mode, you can define the inode number and access control list for the fileset. Additionally, in the *Custom* mode, you can choose whether the new fileset must be added to existing ILM policy rules with fileset scope.

Note: As the *Custom* mode contains more options, this procedure explains how to create a fileset in the *Custom* mode.

4. Click **Browse** to select the path of the junction in the Junction path field. The junction path must be on one of the file systems and it must not refer to any existing file or directory.
5. In the **Name** field, type the name of the fileset.
6. In the **Comment** field, type the comments, if any.
7. Select either **Independent** or **Dependent** from the **Type** field.

An independent fileset has a separate inode space but shares physical storage with the remainder of the file system. Maximum number of inodes and preallocation of inodes for an independent fileset can be specified while creating the fileset. A dependent fileset shares the inode space and snapshot capability of the containing independent fileset.

8. In the **Maximum number of inodes** field, specify the maximum number of file system objects such as files, directories, or links that can be stored under the independent fileset, including that of the related child filesets.
9. In the **Allocated number of inodes** field, specify the number of inodes that is allocated when the fileset is created. The maximum allowed inodes cannot be less than the allocated number.
10. Click **Edit** in the **Edit Action Control** section to define access control list for the users who can access the fileset.

Note: Only NFSv4 ACL semantics are supported in the GUI. The Edit Access Control section is not shown when creating filesets in a file system that supports only POSIX ACL.

11. Select the archive mode to be used from the options that are available under the **Archive Mode** section.

The Archive Mode or the integrated archive mode (IAM) mode provides control to prevent files from being changed or deleted unexpectedly. You can set this option either while creating a fileset or when you modify it. The following options are available to set the archive mode of a fileset. The available values are listed in the order of increasing the level of restriction:

- **Off:** No immutability mode is set and the fileset behaves like a regular fileset. This is the default value.
- **Advisory:** Allows setting retention times and WORM protection, but files can be deleted based on the file permissions.
- **Non-Compliant:** In addition to the restrictions in the advisory mode, files cannot be deleted if the retention time has not expired. However, retention times can be reset, and files can be deleted but not changed.
- **Compliant:** In addition to the restrictions in the non-compliant mode, retention time cannot be reset. When the retention time has expired, files can be deleted but not changed.
- **Compliant Plus:** In addition to the restrictions in the compliant mode, renaming of empty directories is not allowed.

12. Click **Create**.

Fileset is created and linked to the junction path. You can see the newly created fileset in the fileset table.

Deleting a fileset

Filesets are deleted with the `mmdelfileset` command.

There are several notes to keep in mind when deleting filesets:

- The root fileset cannot be deleted.
- A fileset that is not empty cannot be deleted unless the `-f` flag is specified.
- A fileset that is currently linked into the namespace cannot be deleted until it is unlinked with the `mmunlinkfileset` command.
- A dependent fileset can be deleted at any time.
- An independent fileset cannot be deleted if it has any dependent filesets or fileset snapshots.
- Deleting a dependent fileset that is included in a fileset or global snapshot removes it from the active file system, but it remains part of the file system in a deleted state.
- Deleting an independent fileset that is included in any global snapshots removes it from the active file system, but it remains part of the file system in a deleted state.
- A fileset in the deleted state is displayed in the `mmlsfileset` output with the fileset name in parenthesis. If the `-L` flag is specified, the latest including snapshot is also displayed. The `--deleted` option of the `mmlsfileset` command can be used to display only deleted filesets.

- The contents of a deleted fileset are still available in the snapshot, through some path name containing a .snapshots component, because it was saved when the snapshot was created.
- When the last snapshot that includes the fileset has been deleted, the fileset is fully removed from the file system.

For complete usage information, see the topics `mmdelfilesset command`, `mmlsfilesset command`, and `mmunlinkfilesset command` in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related tasks

Creating a fileset

Filesets are created with the `mmcrfilesset` command.

Linking a fileset

After the fileset is created, a junction must be created to link it to the desired location in the file system's namespace using the `mmlinkfilesset` command.

Unlinking a fileset

A junction to a fileset is removed with the `mmunlinkfilesset` command, which unlinks the fileset only from the active directory namespace. The linked or unlinked state of a fileset in a snapshot is unaffected. The unlink fails if there are files open in the fileset, unless the `-f` option is specified. The root fileset cannot be unlinked.

Changing fileset attributes

To change a junction to a fileset, you have to first unlink the fileset using the `mmunlinkfilesset` command, and then create the new junction using the `mmlinkfilesset` command.

Displaying fileset information

Fileset status and attributes are displayed with the `mmlsfilesset` command.

Linking a fileset

After the fileset is created, a junction must be created to link it to the desired location in the file system's namespace using the `mmlinkfilesset` command.

The file system must be mounted in order to link a fileset. An independent fileset can be linked into only one location anywhere in the namespace, specified by the *JunctionPath* parameter:

- The root directory
- Any subdirectory
- The root fileset or to any other fileset

A dependent fileset can only be linked inside its own inode space.

If *JunctionPath* is not specified, the junction is created in the current directory and has the same name as the fileset being linked. After the command completes, the new junction appears as an ordinary directory, except that the user is not allowed to unlink or delete it with the `rmdir` command. The user can use the `mv` command on the directory to move to a new location in the parent fileset, but the `mv` command is not allowed to move the junction to a different fileset.

For complete usage information, see the topic `mmlinkfilesset command` in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related tasks

Creating a fileset

Filesets are created with the `mmcrfilesset` command.

Deleting a fileset

Filesets are deleted with the `mmdelfilesset` command.

Unlinking a fileset

A junction to a fileset is removed with the `mmunlinkfilesset` command, which unlinks the fileset only from the active directory namespace. The linked or unlinked state of a fileset in a snapshot is unaffected.

The unlink fails if there are files open in the fileset, unless the `-f` option is specified. The root fileset cannot be unlinked.

Changing fileset attributes

To change a junction to a fileset, you have to first unlink the fileset using the `mmunlinkfileset` command, and then create the new junction using the `mmlinkfileset` command.

Displaying fileset information

Fileset status and attributes are displayed with the `mmlsfileset` command.

Unlinking a fileset

A junction to a fileset is removed with the `mmunlinkfileset` command, which unlinks the fileset only from the active directory namespace. The linked or unlinked state of a fileset in a snapshot is unaffected. The unlink fails if there are files open in the fileset, unless the `-f` option is specified. The root fileset cannot be unlinked.

After issuing the `mmunlinkfileset` command, the fileset can be re-linked to a different parent using the `mmlinkfileset` command. Until the fileset is re-linked, it is not accessible.

Note: If run against a file system that has an unlinked fileset, `mmapplypolicy` will not traverse the unlinked fileset.

 **Attention:** If you are using the IBM Storage Protect Backup-Archive client you must use caution when you unlink filesets that contain data backed up by IBM Storage Protect. IBM Storage Protect tracks files by pathname and does not track filesets. As a result, when you unlink a fileset, it appears to IBM Storage Protect that you deleted the contents of the fileset. Therefore, the IBM Storage Protect Backup-Archive client inactivates the data on the IBM Storage Protect server which may result in the loss of backup data during the expiration process.

For complete usage information, see the topic *mmunlinkfileset command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related tasks

Creating a fileset

Filesets are created with the `mmcrfileset` command.

Deleting a fileset

Filesets are deleted with the `mmdelfileset` command.

Linking a fileset

After the fileset is created, a junction must be created to link it to the desired location in the file system's namespace using the `mmlinkfileset` command.

Changing fileset attributes

To change a junction to a fileset, you have to first unlink the fileset using the `mmunlinkfileset` command, and then create the new junction using the `mmlinkfileset` command.

Displaying fileset information

Fileset status and attributes are displayed with the `mmlsfileset` command.

Changing fileset attributes

To change a junction to a fileset, you have to first unlink the fileset using the `mmunlinkfileset` command, and then create the new junction using the `mmlinkfileset` command.

To change the attributes of an existing fileset, including the fileset name, use the `mmchfileset` command.

Note: In an HSM-managed file system, moving or renaming migrated files between filesets will result in recalling of the date from the IBM Storage Protect server.

For complete usage information, see the topics *mmchfileset command*, *mmlinkfileset command*, and *mmunlinkfileset command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related tasks

Creating a fileset

Filesets are created with the `mmcrfileset` command.

Deleting a fileset

Filesets are deleted with the `mmdelfileset` command.

Linking a fileset

After the fileset is created, a junction must be created to link it to the desired location in the file system's namespace using the `mmlinkfileset` command.

Unlinking a fileset

A junction to a fileset is removed with the `mmunlinkfileset` command, which unlinks the fileset only from the active directory namespace. The linked or unlinked state of a fileset in a snapshot is unaffected. The unlink fails if there are files open in the fileset, unless the `-f` option is specified. The root fileset cannot be unlinked.

Displaying fileset information

Fileset status and attributes are displayed with the `mmlsfileset` command.

Displaying fileset information

Fileset status and attributes are displayed with the `mmlsfileset` command.

Some of the attributes displayed include:

- Name of the fileset.
- Fileset identifier of the fileset.
- Junction path to the fileset.
- Status of the fileset.
- Root inode number of the fileset.
- Path to the fileset (if linked).
- Inode space.
- User provided comments (if any).

For complete usage information, see the topic *mmlsfileset command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

To display the name of the fileset that includes a given file, run the `mmlsattr` command and specify the `-L` option. For complete usage information, see the topic *mmlsattr command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related tasks

Creating a fileset

Filesets are created with the `mmcrfileset` command.

Deleting a fileset

Filesets are deleted with the `mmdelfileset` command.

Linking a fileset

After the fileset is created, a junction must be created to link it to the desired location in the file system's namespace using the `mmlinkfileset` command.

Unlinking a fileset

A junction to a fileset is removed with the `mmunlinkfileset` command, which unlinks the fileset only from the active directory namespace. The linked or unlinked state of a fileset in a snapshot is unaffected. The unlink fails if there are files open in the fileset, unless the `-f` option is specified. The root fileset cannot be unlinked.

Changing fileset attributes

To change a junction to a fileset, you have to first unlink the fileset using the `mmunlinkfileset` command, and then create the new junction using the `mmlinkfileset` command.

Immutability and appendOnly features

To prevent files from being changed or deleted unexpectedly, GPFS provides immutability and appendOnly restrictions.

Applying immutability and appendOnly restrictions to individual files or to directories

You can apply immutability and appendOnly restrictions either to individual files within a fileset or to a directory.

An immutable file cannot be changed or renamed. An appendOnly file allows append operations, but not delete, modify, or rename operations.

An immutable directory cannot be deleted or renamed, and files cannot be added or deleted under such a directory. An appendOnly directory allows new files or subdirectories to be created with 0 byte length; all such new created files and subdirectories are marked as appendOnly automatically.

The `immutable` flag and the `appendOnly` flag can be set independently. If both immutability and appendOnly are set on a file, immutability restrictions will be in effect.

To set or unset these attributes, use the following command options:

mmchattr -i {yes | no}

Sets or unsets a file to or from an immutable state.

-i yes

Sets the `immutable` attribute of the file to yes.

-i no

Sets the `immutable` attribute of the file to no.

mmchattr -a {yes | no}

Sets or unsets a file to or from an appendOnly state.

-a yes

Sets the `appendOnly` attribute of the file to yes.

-a no

Sets the `appendOnly` attribute of the file to no.

Note: Before an immutable or appendOnly file can be deleted, you must change it to mutable or set `appendOnly` to no (by using the `mmchattr` command).

Storage pool assignment of an immutable or appendOnly file can be changed; an immutable or appendOnly file is allowed to transfer from one storage pool to another.

To display whether or not a file is immutable or appendOnly, issue this command:

```
mmclsattr -L myfile
```

The system displays information similar to the following:

```
file name:          myfile
metadata replication: 2 max 2
data replication:   1 max 2
immutable:          no
appendOnly:         no
flags:
storage pool name: sp1
fileset name:       root
snapshot name:
```

The effects of file operations on immutable and appendOnly files

Once a file has been set as immutable or appendOnly, the following file operations and attributes work differently from the way they work on regular files:

delete

An immutable or appendOnly file cannot be deleted.

modify/append

An appendOnly file cannot be modified, but it can be appended. An immutable file cannot be modified or appended.

Note: The immutable and appendOnly flag check takes effect after the file is closed; therefore, the file can be modified if it is opened before the file is changed to immutable.

mode

An immutable or appendOnly file's mode cannot be changed.

ownership, acl

These attributes cannot be changed for an immutable or appendOnly file.

extended attributes

These attributes cannot be added, deleted, or modified for an immutable or appendOnly file.

timestamp

The timestamp of an immutable or appendOnly file can be changed.

directory

If a directory is marked as immutable, no files can be created, renamed, or deleted under that directory. However, a subdirectory under an immutable directory remains mutable unless it is explicitly changed by mmchattr.

If a directory is marked as appendOnly, no files can be renamed or deleted under that directory. However, 0 byte length files can be created.

The following table shows the effects of file operations on an immutable file or an appendOnly file.

Table 46. The effects of file operations on an immutable file or an appendOnly file		
Operation	immutable	appendOnly
Add, delete, modify, or rename	No	No
Append	No	Yes
Change ownership, mode, or acl	No	No
Change atime, mtime, or ctime	Yes	Yes
Add, delete, or modify extended attributes	Disallowed by external methods such as setfattr. Allowed internally for dmapi, directio, and others.	Same as for immutable.
Create a file under an immutable or appendOnly directory	No	Yes, 0 byte length only
Rename or delete a file under an immutable or appendOnly directory	No	No
Modify a mutable file under an immutable directory	Yes	Not applicable

Table 46. The effects of file operations on an immutable file or an appendOnly file (continued)

Operation	immutable	appendOnly
Set an immutable file back to mutable	Yes	Not applicable
Set an appendOnly file back to a non-appendOnly state	Not applicable	Yes

Fileset-level integrated archive manager (IAM) modes

You can modify the file-operation restrictions that apply to the immutable files in a fileset by setting an integrated archive manager (IAM) mode for the fileset. The following table shows the effects of each of the IAM modes.

Note: To set an IAM mode for a fileset, issue the **mmchfileset** command with the --iam-mode parameter. For more information, see the topic *mmchfileset* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Table 47. IAM modes and their effects on file operations on immutable files

File operation	Regular mode	Advisory mode	Noncompliant mode	Compliant mode	Compliant-plus mode
Modify	No	No	No	No	No
Append	No	No	No	No	No
Rename	No	No	No	No	No
Change ownership, acl	No	No	No	No	No
Change mode	No	No	No	No	No
Change atime, mtime, ctime	Yes	mtime and ctime can be changed. atime is overloaded by expiration time. Expiration time can be changed by using the mmchattr --expiration-time command (alternatively mmchattr -E) or touch. You can see the expiration time by using stat as atime.	Same as advisory mode	Same as advisory mode	Same as advisory mode

Table 47. IAM modes and their effects on file operations on immutable files (continued)

File operation	Regular mode	Advisory mode	Noncompliant mode	Compliant mode	Compliant-plus mode
Add, delete, or modify extended attributes.	Not allowed for external methods such as <code>setfattr</code> . Allowed internally for dmapi, directio, and etc.	Yes	Yes	Yes	Yes
Create, rename, or delete under an immutable directory	No	No	No	No	No
Modify mutable files under an immutable directory.	Yes	Yes	Yes	Yes	Yes
Retention rule enforced	No retention rule, cannot delete immutable files	No	Yes	Yes	Yes
Set ExpirationTime backwards	Yes	Yes	Yes	No	No
Delete an immutable file	No	Yes, always	Yes, only when expired	Yes, only when expired	Yes, only when expired
Set an immutable file back to mutable	Yes	No	No	No	No
Allow hardlink	No for immutable or appendOnly files. Yes for other files.	No	No	No	No
Rename or delete a non-empty directory	Yes for rename. No for delete only if the directory contains immutable files.	No for rename. Yes for delete.	No for rename. Yes for delete only if the immutable file has expired.	No for rename. Yes for delete only if the immutable file has expired.	No for rename. Yes for delete only if the immutable file has expired.
Rename an empty directory	Yes	Yes	Yes	Yes	No

Table 47. IAM modes and their effects on file operations on immutable files (continued)

File operation	Regular mode	Advisory mode	Noncompliant mode	Compliant mode	Compliant-plus mode
Remove user write permission to change a file to immutable	No	Yes	Yes	Yes	Yes
Display expiration time instead of atime for stat call	No	Yes	Yes	Yes	Yes
Set a directory to be immutable	Yes	No	No	No	No

Creating and applying ILM policy by using GUI

The Information Lifecycle Management (ILM) feature that is available in the IBM Storage Scale system facilitates automated tiered storage management. You need to create a set of policies and rules that automatically determine where to physically store your data regardless of its placement in the logical directory structure. Proper management of files ensures the efficient use and balance of premium and less expensive storage resources.

You can use the **Files > Information Lifecycle** page in the IBM Storage Scale GUI to create and manage the ILM policies. The Information Lifecycle page consists of the following tabs:

- **Active Policy:** Lists the policy that is applied to the file systems. You can add new rules or run policy from this section. A default placement policy is added to the file system if the file system is created by using the GUI.
- **Policy Repository:** Repository of the policies that are configured in the system. You can create new policy and add it in to the repository or make an existing policy as the active policy for a file system. You can even run a policy from the Policy Repository section without making it as the active policy.
- **Policy Run Settings:** Provides the options to select the nodes that run the policy, select temporary folder to be used while running the policy, and configure certain performance tuning.

Defining and applying an ILM policy includes the following steps at a high-level:

1. Set policy run settings.
2. If the file system is not created through GUI, create a policy and make it as the active policy for the file system.
3. If the file system is created by using GUI, either modify the default placement policy based on your requirement or create a policy and apply it as the active policy.

Perform the following steps to create and apply an ILM policy:

1. Go to **Files > Information Lifecycle** page in the IBM Storage Scale GUI. The Information Lifecycle page appears.
2. Select **Policy Run Settings**.
3. In the **Nodes that run policies** field, select the criterion for which the node or nodes to be selected.

You can select the following values:

- Master nodes
- Node class

- Individual nodes
- If you select **Node Class** or **Individual Nodes** as the criterion, you need to specify the node classes and nodes.
 - In the **Local work directory** field, specify the local directory to be selected for temporary storage.
 - In the **Global work directory** field, specify a global work directory for the temporary storage, if you want to use a global work directory instead of a local work directory.
 - Specify the following performance tuning parameters:
 - Average number of CPU cores per node
 - Number of threads for policy scan
 - Number of threads for policy execution
 - Click **Save** to save the changes that are made to the policy run settings.

Note: Assuming that the file system is not created through GUI. Hence, this procedure explains the steps to create a policy and make it as the active policy for the file system. If the file system is created by using GUI, either modify the default placement policy based on your requirement or create a policy and apply it as the active policy.

- Click **Policy Repository**.
 - Click the add symbol or select **Create Policy** from the **Actions** menu. The Create Policy window appears.
 - In the Create Policy window, specify the policy name and file system for which the policy is applicable.
 - Click **Create** to create the policy.
- The policy is created. Now, you need to add rules in the policy that manages the files in the system.
- Click **Add Rule** in the Policy Repository and define rules with the required rule types. The **Add Rule** option only supports to add placement, migration, file compression, encryption, exclusion, or deletion rules, or to define an external pool. To add list rules, the policy text must be modified by using the text editor.
 - You can create multiple rules in a policy. You can drag the rules in the rules list to change the order in which the rules are applied in a policy.
 - Optionally, you can use the text editor to edit policy text. Click **Policy Text** option that is available in the upper right corner of the GUI page to launch the text editor. The support for expressions are also more in the text editor. The list rules are supported only on the text editor. After editing the policy details, click **Apply Changes**.
 - Select the policy from the **Policy Repository** and then select **Apply as Active Policy** option that is available in the **Actions** menu. You can also change the active policy of the file system.

Note: The GUI does not support scheduling of policy runs. Not all tuning options are available in the Policy Run Settings. Some rule types and complex expressions are also not supported. Therefore, you need to use the **mmapplypolicy** command to support specific ILM related actions.

Modifying active ILM policy by using GUI

You can use the **Files > Information Lifecycle** page in the GUI to create and manage the ILM policies.

Perform the following steps to create and apply an ILM policy:

- Go to **Files > Information Lifecycle** page in the IBM Storage Scale GUI.
- Click **Active Policy**.
- In the **File System** field, select the file system for which you need to modify the active policy.
- Review the policy details. The available rules are listed on the left side of the view and the selected rule details is displayed on the right side of the page.
- Click **Add Rule** to add new rules to the policy. The Add Rule window appears.
- In the Add Rule window, specify the rule name and rule type. The following rule types can be selected:

- Migration
- Migration to external pool
- Placement
- Compression
- Migration and compression
- Deletion
- Exclude
- Encryption
- Encryption specification
- Encryption exclude
- External pool

7. Click **Add** to add the rule to the policy. The rule is added to the policy.
8. Select the rule from the list of rules that are added to the policy. The rule definition appears on the right pane.
9. Modify the default values of the rule based on the requirement.
10. Click **Apply Changes** after you make the required changes.
11. To remove a rule from an active policy, select the rule from the list of rules that are configured for the policy, and clear the **Enable** checkbox from the right pane. Make sure that you click **Apply Changes** whenever you update rules of a policy.
12. Click **Run Policy** if you want to run the policy irrespective of the conditions specified in the rules of the policy. Usually, the system runs the policy when the conditions mentioned in the rules are met. For example, migration starts when the thresholds that are defined in the migration rule is reached.

Chapter 40. Creating and maintaining snapshots of file systems

A snapshot of an entire GPFS file system can be created to preserve the contents of the file system at a single point in time. Snapshots of the entire file system are also known as global snapshots. The storage overhead for maintaining a snapshot is keeping a copy of data blocks that would otherwise be changed or deleted after the time of the snapshot.

Snapshots of a file system are read-only; changes can only be made to the active (that is, normal, non-snapshot) files and directories.

The snapshot function allows a backup or mirror program to run concurrently with user updates and still obtain a consistent copy of the file system as of the time that the snapshot was created. Snapshots also provide an online backup capability that allows easy recovery from common problems such as accidental deletion of a file, and comparison with older versions of a file.

Notes:

1. Because snapshots are not copies of the entire file system, they should not be used as protection against media failures. For information about protection against media failures, see the topic *Recoverability considerations in the IBM Storage Scale: Concepts, Planning, and Installation Guide*.
2. Fileset snapshots provide a method to create a snapshot of an independent fileset instead of the entire file system. For more information about fileset snapshots, see [“Fileset-level snapshots” on page 586](#).
3. A snapshot of a file creates a new file that captures the user data and user attributes from the original. The snapshot file is independent from the original file. For DMAPI managed file systems, the snapshot of a file is not automatically managed by DMAPI, regardless of the state of the original file. The DMAPI attributes from the original file are not inherited by the snapshot. For more information about DMAPI restrictions for GPFS, see the *IBM Storage Scale: Command and Programming Reference Guide*.
4. When snapshots are present, deleting files from the active file system does not always result in any space actually being freed up; rather, blocks may be pushed to the previous snapshot. In this situation, the way to free up space is to delete the oldest snapshot. Before creating new snapshots, it is good practice to ensure that the file system is not close to being full.
5. The use of clones functionally provides writable snapshots. See [Chapter 41, “Creating and managing file clones,” on page 613](#).

Management of snapshots of a GPFS file system includes:

- [“Creating a snapshot” on page 601](#)
- [“Listing snapshots” on page 603](#)
- [“Restoring a file system from a snapshot” on page 604](#)
- [“Reading a snapshot with the policy engine” on page 605](#)
- [“Linking to a snapshot” on page 606](#)
- [“Deleting a snapshot” on page 607](#)

Creating a snapshot

Use the `mmcrsnapshot` command to create a snapshot of an entire GPFS file system at a single point in time. Snapshots appear in the file system tree as hidden subdirectories of the root.

Global snapshots appear in a subdirectory in the root directory of the file system, whose default name is `.snapshots`. If you prefer to access snapshots from each directory rather than traversing through the root directory, then you can use an invisible directory to make the connection by issuing the `mmsnapdir` command. For more information, see [“Linking to a snapshot” on page 606](#).

A snapshot of the file system, *Device*, is identified by a SnapshotName name on the **mmcrsnapshot** command. For example, given the file system *fs1* to create a snapshot *snap1*, enter:

```
mmcrsnapshot fs1 snap1
```

The output is similar to this:

```
Writing dirty data to disk.  
Quiescing all file system operations.  
Writing dirty data to disk again.  
Snapshot snap1 created with id 1.
```

Before issuing the command, the directory structure would appear similar to:

```
/fs1/file1  
/fs1/userA/file2  
/fs1/userA/file3
```

After the command is issued, the directory structure would appear similar to:

```
/fs1/file1  
/fs1/userA/file2  
/fs1/userA/file3  
  
/fs1/.snapshots/snap1/file1  
/fs1/.snapshots/snap1/userA/file2  
/fs1/.snapshots/snap1/userA/file3
```

If a second snapshot is supposed to be created later, then the first snapshot would remain as it is. A snapshot can be made only of an active file system, not of an existing snapshot. The following command creates another snapshot of the same file system:

```
mmcrsnapshot fs1 snap2
```

The output is similar to this:

```
Writing dirty data to disk.  
Quiescing all file system operations.  
Writing dirty data to disk again.  
Snapshot snap2 created with id 2.
```

After the command is issued, the directory structure would appear similar to:

```
/fs1/file1  
/fs1/userA/file2  
/fs1/userA/file3  
  
/fs1/.snapshots/snap1/file1  
/fs1/.snapshots/snap1/userA/file2  
/fs1/.snapshots/snap1/userA/file3  
  
/fs1/.snapshots/snap2/file1  
/fs1/.snapshots/snap2/userA/file2  
/fs1/.snapshots/snap2/userA/file3
```

For more information, see the *mmcrsnapshot* command in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

[Reading a snapshot with the policy engine](#)

[Managing snapshots using IBM Storage Scale GUI](#)

Use **Files > Snapshots** page in the IBM Storage Scale GUI to manage snapshots through GUI.

Related tasks

[Listing snapshots](#)

Use the **mmlssnapshot** command to display existing snapshots of a file system and their attributes.

[Restoring a file system from a snapshot](#)

Use the `mmrestorefs` command to restore user data and attribute files in an active file system from a snapshot.

Linking to a snapshot

Snapshot root directories appear in a special `.snapshots` directory under the file system root.

Deleting a snapshot

Use the `mmdelsnapshot` command to delete GPFS snapshots of a file system.

Creating a snapshot by using GUI

A snapshot can be created manually or based on a schedule by defining the snapshot rules. You can also create an AFM peer snapshot from the Snapshots page. The following procedure explains how to schedule snapshot creation and retention by defining snapshot rules.

Perform the following steps to create a snapshot:

1. Go to **Files > Snapshots** page in the IBM Storage Scale GUI.
2. Click **Create Snapshot**. The Create Snapshot window appears.

The Create Snapshot window by default shows options to create snapshots under the manual mode. If you want to schedule snapshot creation of a file system or an independent fileset, use the options that are provided under the **Schedule** tab of the Create Snapshot window. This procedure shows how to schedule a snapshot creation.

3. Click **Schedule**.
4. In the **Path** field, type the path of the file system or independent fileset for which you need to create snapshots.
5. Click **Create Rule** to schedule the snapshot creation and retention. Create Snapshot Rule window appears.
6. In the **Name** field, type the name of the snapshot scheduling rule.
7. In the **Frequency** field, select the frequency in which you need to create snapshot. You need to enter some more details based on the value that is selected in the Frequency field. For example, if value selected is *Multiple Times an Hour*, select the minutes of the hour in which you need to create snapshots.
8. In the **Retention** fields, specify the number of snapshots that must be retained in a time period.
9. In the **Prefix** field, specify a prefix to be added with the name of the snapshots that are created with this rule. The prefix is added to the date and time to identify the rule that is used to create the snapshot. If a prefix is not specified, the default prefix @GMT is used. Using the default prefix enables Microsoft Windows Volume Shadow Copy Service (VSS) identification if the file is shared by using the SMB protocol.
10. Select the **Allow Expiration** checkbox to delete the snapshot when the defined retention period is completed.
Note: If you do not select the **Allow Expiration** checkbox then the expiration time for all snapshots created is the same as its creation time.
11. Click **OK** to create the snapshot rule. You can associate more than one rule to a path.
12. In the **Create Snapshot** window, click **Create** to create the snapshot creation schedule and retention policy for the snapshots. The snapshot of the file system or independent fileset will be created based on the schedule specified in the snapshot rule.

Listing snapshots

Use the `mmlssnapshot` command to display existing snapshots of a file system and their attributes.

The `-d` option displays the amount of storage used by a snapshot. GPFS quota management does not take the data blocks used to store snapshots into account when reporting on and determining if quota limits have been exceeded. This is a slow operation and its usage is suggested for problem determination only.

For example, to display the snapshot information for the file system `fs1` with additional storage information, issue the following command:

```
mmlssnapshot fs1 -d
```

The system displays information similar to:

Snapshots in file system <code>fs1</code> : [data and metadata in KB]				Data	Metadata
Directory	SnapId	Status	Created		
<code>snap1</code>	1	Valid	Fri Oct 17 10:56:22 2003	0	512

For complete usage information, see the topic *mmlssnapshot command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

[Reading a snapshot with the policy engine](#)

[Managing snapshots using IBM Storage Scale GUI](#)

Use **Files > Snapshots** page in the IBM Storage Scale GUI to manage snapshots through GUI.

Related tasks

[Creating a snapshot](#)

Use the `mmcrsnapshot` command to create a snapshot of an entire GPFS file system at a single point in time. Snapshots appear in the file system tree as hidden subdirectories of the root.

[Restoring a file system from a snapshot](#)

Use the `mmrestorefs` command to restore user data and attribute files in an active file system from a snapshot.

[Linking to a snapshot](#)

Snapshot root directories appear in a special `.snapshots` directory under the file system root.

[Deleting a snapshot](#)

Use the `mmdelsnapshot` command to delete GPFS snapshots of a file system.

Restoring a file system from a snapshot

Use the `mmrestorefs` command to restore user data and attribute files in an active file system from a snapshot.

Prior to issuing the `mmrestorefs` command, ensure that the file system is mounted. When restoring from an independent fileset snapshot, ensure that the fileset is in linked state.

Existing snapshots, including the one being used in the restore, are not modified by the `mmrestorefs` command. To obtain a snapshot of the restored file system, you must issue the `mmcrsnapshot` command to capture it before issuing the `mmrestorefs` command again.

As an example, suppose that you have a directory structure similar to the following:

```
/fs1/file1  
/fs1/userA/file2  
/fs1/userA/file3  
/fs1/.snapshots/snap1/file1  
/fs1/.snapshots/snap1/userA/file2  
/fs1/.snapshots/snap1/userA/file3
```

If the directory `userA` is then deleted, the structure becomes similar to this:

```
/fs1/file1  
/fs1/.snapshots/snap1/file1  
/fs1/.snapshots/snap1/userA/file2  
/fs1/.snapshots/snap1/userA/file3
```

The directory `userB` is then created using the inode originally assigned to `userA`, and another snapshot is taken:

```
mmcrsnapshot fs1 snap2
```

The output is similar to this:

```
Writing dirty data to disk.  
Quiescing all file system operations.  
Writing dirty data to disk again.  
Snapshot snap2 created with id 2.
```

The resulting directory structure is similar to the following:

```
/fs1/file1  
/fs1/userB/file2b  
/fs1/userB/file3b  
/fs1/.snapshots/snap1/file1  
/fs1/.snapshots/snap1/userA/file2  
/fs1/.snapshots/snap1/userA/file3  
/fs1/.snapshots/snap2/file1  
/fs1/.snapshots/snap2/userB/file2b  
/fs1/.snapshots/snap2/userB/file3b
```

The file system is then restored from `snap1`:

```
mmrestorefs fs1 snap1
```

The resulting directory structure is similar to the following:

```
/fs1/file1  
/fs1/userA/file2  
/fs1/userA/file3  
/fs1/.snapshots/snap1/file1  
/fs1/.snapshots/snap1/userA/file2  
/fs1/.snapshots/snap1/userA/file3  
/fs1/.snapshots/snap2/file1  
/fs1/.snapshots/snap2/userB/file2b  
/fs1/.snapshots/snap2/userB/file3b
```

For complete usage information, see the topic *mmrestorefs command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

[Reading a snapshot with the policy engine](#)

[Managing snapshots using IBM Storage Scale GUI](#)

Use **Files > Snapshots** page in the IBM Storage Scale GUI to manage snapshots through GUI.

Related tasks

[Creating a snapshot](#)

Use the `mmcrsnapshot` command to create a snapshot of an entire GPFS file system at a single point in time. Snapshots appear in the file system tree as hidden subdirectories of the root.

[Listing snapshots](#)

Use the `mmlssnapshot` command to display existing snapshots of a file system and their attributes.

[Linking to a snapshot](#)

Snapshot root directories appear in a special `.snapshots` directory under the file system root.

[Deleting a snapshot](#)

Use the `mmdelsnapshot` command to delete GPFS snapshots of a file system.

Reading a snapshot with the policy engine

You can use the policy engine to read the contents of a snapshot for backup purposes. The `mmaplypolicy` command provides the `-S` option to specify the snapshot during a policy run. Instead of matching rules to the active file system, the policy engine matches the rules against files in the snapshot.

Notes:

1. Snapshots are read-only. Policy rules such as MIGRATE or DELETE that make changes or delete files cannot be used with a snapshot.
2. An instance of `mmapplypolicy` can only scan one snapshot. Directing it at the `.snapshots` directory itself will result in a failure.

For complete usage information, see the topic *mmapplypolicy command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

[Managing snapshots using IBM Storage Scale GUI](#)

Use **Files > Snapshots** page in the IBM Storage Scale GUI to manage snapshots through GUI.

Related tasks

[Creating a snapshot](#)

Use the `mmcirsnapshot` command to create a snapshot of an entire GPFS file system at a single point in time. Snapshots appear in the file system tree as hidden subdirectories of the root.

[Listing snapshots](#)

Use the `mmlsssnapshot` command to display existing snapshots of a file system and their attributes.

[Restoring a file system from a snapshot](#)

Use the `mmrestoresfs` command to restore user data and attribute files in an active file system from a snapshot.

[Linking to a snapshot](#)

Snapshot root directories appear in a special `.snapshots` directory under the file system root.

[Deleting a snapshot](#)

Use the `mmdelssnapshot` command to delete GPFS snapshots of a file system.

Linking to a snapshot

Snapshot root directories appear in a special `.snapshots` directory under the file system root.

If you prefer to link directly to the snapshot rather than always traverse the root directory, you can use the `mmsnapdir` command with the `-a` option to add a `.snapshots` subdirectory to all directories in the file system. These `.snapshots` subdirectories will contain a link into the corresponding directory for each snapshot that includes the directory in the active file system.

Unlike `.snapshots` in the root directory, however, the `.snapshots` directories added by the `-a` option of the `mmsnapdir` command are invisible in the sense that the `ls` command or `readdir()` function does not return `.snapshots`. This is to prevent recursive file system utilities such as `find` or `tar` from entering into the snapshot tree for each directory they process. For example, if you enter `ls -a /fs1/userA`, the `.snapshots` directory is not listed. However, you can enter `ls /fs1/userA/.snapshots` or `cd /fs1/userA/.snapshots` to confirm that `.snapshots` is present. If a user wants to make one of their snapshot directories more visible, it is suggested to create a symbolic link to `.snapshots`.

The inode numbers that are used for and within these special `.snapshots` directories are constructed dynamically and do not follow the standard rules. These inode numbers are visible to applications through standard commands, such as `stat`, `readdir`, or `ls`. The inode numbers reported for these directories can also be reported differently on different operating systems. Applications should not expect consistent numbering for such inodes.

Specifying the `-r` option on the `mmsnapdir` command reverses the effect of the `-a` option, and reverts to the default behavior of a single `.snapshots` directory in the root directory.

The `-s` option allows you to change the name of the `.snapshots` directory. For complete usage information, see the topic *mmsnapdir command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

To illustrate this point, assume that a GPFS file system called **fs1**, which is mounted at `/fs1`, has one snapshot called **snap1**. The file system might appear similar to this:

```
/fs1/userA/file2b  
/fs1/userA/file3b  
/fs1/.snapshots/snap1/userA/file2b  
/fs1/.snapshots/snap1/userA/file3b
```

To create links to the snapshots from each directory, and instead of `.snapshots`, use the name `.links`, enter:

```
mmsnapdir fs1 -a -s .links
```

After the command completes, the directory structure would appear similar to:

```
/fs1/userA/file2b  
/fs1/userA/file3b  
/fs1/userA/.links/snap1/file2b  
/fs1/userA/.links/snap1/file3b  
  
/fs1/.links/snap1/userA/file2b  
/fs1/.links/snap1/userA/file3b
```

To delete the links, issue:

```
mmsnapdir fs1 -r
```

After the command completes, the directory structure is similar to the following:

```
/fs1/userA/file2b  
/fs1/userA/file3b  
  
/fs1/.links/snap1/userA/file2b  
/fs1/.links/snap1/userA/file3b
```

For complete usage information, see the topic *mmsnapdir command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

[Reading a snapshot with the policy engine](#)

[Managing snapshots using IBM Storage Scale GUI](#)

Use **Files > Snapshots** page in the IBM Storage Scale GUI to manage snapshots through GUI.

Related tasks

[Creating a snapshot](#)

Use the `mmcrsnapshot` command to create a snapshot of an entire GPFS file system at a single point in time. Snapshots appear in the file system tree as hidden subdirectories of the root.

[Listing snapshots](#)

Use the `mmlssnapshot` command to display existing snapshots of a file system and their attributes.

[Restoring a file system from a snapshot](#)

Use the `mmrestorefs` command to restore user data and attribute files in an active file system from a snapshot.

[Deleting a snapshot](#)

Use the `mmdelsnapshot` command to delete GPFS snapshots of a file system.

Deleting a snapshot

Use the `mmdelsnapshot` command to delete GPFS snapshots of a file system.

For example, to delete **snap1** for the file system **fs1**, enter:

```
mmdelsnapshot fs1 snap1
```

The output is similar to this:

```
Invalidate snapshot files...
Deleting snapshot files...
100.00 % complete on Tue Feb 28 10:40:59 2012
Delete snapshot snap1 complete, err = 0
```

For complete usage information, see the topic *mmdelsnapshot command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Related concepts

[Reading a snapshot with the policy engine](#)

[Managing snapshots using IBM Storage Scale GUI](#)

Use **Files > Snapshots** page in the IBM Storage Scale GUI to manage snapshots through GUI.

Related tasks

[Creating a snapshot](#)

Use the **mmcrsnapshot** command to create a snapshot of an entire GPFS file system at a single point in time. Snapshots appear in the file system tree as hidden subdirectories of the root.

[Listing snapshots](#)

Use the **mmlssnapshot** command to display existing snapshots of a file system and their attributes.

[Restoring a file system from a snapshot](#)

Use the **mmrestoresfs** command to restore user data and attribute files in an active file system from a snapshot.

[Linking to a snapshot](#)

Snapshot root directories appear in a special `.snapshots` directory under the file system root.

Managing snapshots using IBM Storage Scale GUI

Use **Files > Snapshots** page in the IBM Storage Scale GUI to manage snapshots through GUI.

Snapshots can be used in environments where multiple recovery points are necessary. A snapshot can be taken of file system or fileset data and then the data can be recovered from the snapshot if the production data is not available.

Note:

- Snapshots are read-only; changes can be made only to the normal and active files and directories, not to the snapshot.
- When a snapshot of an independent fileset is taken, only nested dependent filesets are included in the snapshot.

Scheduling snapshot creation by using snapshot rules

You can either manually create the snapshots or create snapshot rules to automate the snapshot creation and retention through the IBM Storage Scale GUI. You cannot schedule snapshot creation if you create snapshots by using the **mmcrsnapshot** CLI command. However, you can use the GUI CLI command options that are available at the following directory on the GUI node to schedule snapshot creation: `/usr/lpp/mmfs/gui/cli`.

To manually create a snapshot, click **Create Snapshot** in the **Snapshots** page and enter the necessary details under the **Manual** tab of the **Create Snapshot** window. Click **Create** after entering the details.

By creating a snapshot rule, you can automate the snapshot creation and retention. That is, in a snapshot rule you can specify a frequency in which the snapshots must be created and the number of snapshots that must be retained for a period. The retention policy helps to avoid unwanted storage of snapshots that result in waste of storage resources.

Retention policy has the following parameters:

- Frequency of snapshot creation

- Number of most recent snapshots to be retained. The most recent snapshot is identified based on the frequency of snapshot creation.
- Number of days for which you need to keep the latest snapshot of each day.
- Number of weeks for which you need to keep the latest snapshot of each week.
- Number of months for which you need to keep the latest snapshot of each month.

Example scenario for retention policy

The following table provides an example for the values that are specified against these parameters.

Table 48. Example for retention period						
Frequency	Minute	Number of most recent snapshots	Keep latest snapshots for			
			Hours	Days	Weeks	Months
Hourly	1	2	2	6	2	3

Based on this retention rule, the following snapshots are created and retained on 20 March 2016 at 06:10 AM.

Table 49. Example - Time stamp of snapshots that are retained based on the retention policy	
Time stamp	Condition based on which snapshot is retained
December 31 (Thursday, 11:01 PM)	Keep latest snapshot for last 3 months.
January 31 (Sunday, 11:01 PM)	Keep latest snapshot for last 3 months.
February 29 (Monday, 11:01 PM)	Keep latest snapshot for last 3 months.
March 5 (Saturday, 11:01 PM)	Keep latest snapshot for last 2 weeks.
March 12 (Saturday, 11:01 PM)	Keep latest snapshot for last 2 weeks.
March 14 (Monday, 11:01 PM)	Keep latest snapshot for last 6 days.
March 15 (Tuesday, 11:01 PM)	Keep latest snapshot for last 6 days.
March 16 (Wednesday, 11:01 PM)	Keep latest snapshot for last 6 days.
March 17 (Thursday, 11:01 PM)	Keep latest snapshot for last 6 days.
March 18 (Friday, 11:01 PM)	Keep latest snapshot for last 6 days.
March 19 (Saturday, 11:01 PM)	Keep latest snapshot for last 6 days.
March 20 (Sunday, 5: 01 AM)	Keep 2 most recent.
March 20 (Sunday, 6: 01 AM)	Keep 2 most recent.

According to this rule, 13 snapshots are retained on 20 March 2016 at 06:10 AM.

To schedule snapshot creation and retention, follow the procedure that is shown:

1. Go to **Files > Snapshots**.
2. Click **Create Snapshot**.
3. In the **Create Snapshot** window, enter the path of the file system or independent fileset for which you need to create snapshots.
4. In the **Snapshot name** field, specify the name of the snapshot.
5. Click **Snapshot Rules**.
6. Click **Create Rule** to schedule the snapshot creation and retention. **Create Snapshot Rule** window is displayed.
7. In the **Name** field, type the name of the snapshot scheduling rule.

8. In the **Frequency** field, select the frequency in which you need to create snapshot. You need to enter some more details based on the value that is selected in the **Frequency** field. For example, if value selected is **Multiple Times an Hour**, select the minutes of the hour in which you need to create snapshots.
 9. In the **Retention** fields, specify the number of snapshots that must be retained in a time period.
 10. In the **Prefix** field, specify a prefix to be added with the name of the snapshots that are created with this rule.
 11. Select the **Allow Expiration** checkbox to delete the snapshot when the defined retention period is completed.
- Note:** If you do not select the **Allow Expiration** checkbox then the expiration time for all snapshots created is the same as its creation time.
12. Click **OK** to save the changes.

If you do not specify a name for the snapshot, the default name is given. The default snapshot ID is generated at the creation time by using the format "@*GMT*-*yyyy.MM.dd-HH.mm.ss*". If this option is given and the "@*GMT-date-time*" format is omitted, then this snapshot is not identified by Windows VSS and the file restore is not possible by that method. Avoid white spaces, double and single quotation marks, the parentheses (), the star *, forward slash /, and backward slash \.

Deleting snapshots

To manually delete the snapshots, right-click the snapshot from the **Snapshots** page and select **Delete**. The snapshots that are automatically created based on the snapshot creation rule, are deleted automatically based on the retention period that is specified in the rule. When the condition for deletion is met, the GUI immediately starts to delete the snapshot candidates.

Note: Default snapshots for which retention period is not defined can be deleted any time after their creation.

Creating and deleting peer and RPO snapshots

The peer and recovery point objective (RPO) snapshots are used in the AFM and AFM DR configurations to ensure data integrity and availability. When a peer snapshot is taken, it creates a snapshot of the cache fileset and then queues a snapshot creation at the home site. This process ensures application consistency at both cache and home sites. The RPO snapshot is a type of peer snapshot that is used in the AFM DR setup. It is used to maintain consistency between the primary and secondary sites in an AFM DR configuration.

Use the **Create Peer Snapshot** option in the **Files > Snapshots** page to create peer snapshots. You can view and delete these peer snapshots from the **Snapshots** page and also from the detailed view of the **Files > Active File Management** page.

Related concepts

[Reading a snapshot with the policy engine](#)

Related tasks

[Creating a snapshot](#)

Use the `mmcrsnapshot` command to create a snapshot of an entire GPFS file system at a single point in time. Snapshots appear in the file system tree as hidden subdirectories of the root.

[Listing snapshots](#)

Use the `mmlssnapshot` command to display existing snapshots of a file system and their attributes.

[Restoring a file system from a snapshot](#)

Use the `mmrestorefs` command to restore user data and attribute files in an active file system from a snapshot.

[Linking to a snapshot](#)

Snapshot root directories appear in a special `.snapshots` directory under the file system root.

Deleting a snapshot

Use the `mmdelsnapshot` command to delete GPFS snapshots of a file system.

Chapter 41. Creating and managing file clones

A file clone is a writable snapshot of an individual file. File clones can be used to provision virtual machines by creating a virtual disk for each machine by cloning a common base image. A related usage is to clone the virtual disk image of an individual machine as part of taking a snapshot of the machine state.

Cloning a file is similar to creating a copy of a file, but the creation process is faster and more space efficient because no additional disk space is consumed until the clone or the original file is modified. Multiple clones of the same file can be created with no additional space overhead. You can also create clones of clones.

Management of file clones in a GPFS file system includes:

- [“Creating file clones” on page 613](#)
- [“Listing file clones” on page 614](#)
- [“Deleting file clones” on page 615](#)
- [“File clones and disk space management” on page 615](#)
- [“File clones and snapshots” on page 615](#)

Creating file clones

File clones can be created from a regular file or a file in a snapshot using the `mmclone` command.

Creating a file clone from a regular file is a two-step process using the `mmclone` command with the `snap` and `copy` keywords:

1. Issue the `mmclone snap` command to create a read-only snapshot of the file to be cloned. This read-only snapshot becomes known as the clone parent. For example, the following command creates a clone parent called `snap1` from the original file `file1`:

```
mmclone snap file1 snap1
```

Alternately, if only one file is specified with the `mmclone snap` command, it will convert the file to a read-only clone parent without creating a separate clone parent file. When using this method to create a clone parent, the specified file cannot be open for writing or have hard links. For example, the following command converts `file1` into a clone parent.

```
mmclone snap file1
```

2. Issue the `mmclone copy` command to create a writable clone from a clone parent. For example, the following command creates a writable file clone called `file2` from the clone parent `snap1`:

```
mmclone copy snap1 file2
```

Creating a file clone where the source is in a snapshot only requires one step using the `mmclone` command with the `copy` keyword. For example, the following command creates a writable file clone called `file3.clone` from a file called `file3` in a snapshot called `snap2`:

```
mmclone copy /fs1/.snapshots/snap2/file3 file3.clone
```

In this case, `file3` becomes the clone parent.

Note: Extended attributes of clone parents are not passed along to file clones.

After a clone has been created, the clone and the file that it was cloned from are interchangeable, which is similar to a regular copy (`cp`) command. The file clone will have a new inode number and attributes that can be modified independently of the original file.

Additional clones can be created from the same clone parent by issuing additional `mmclone copy` commands, for example:

```
mmclone copy snap1 file3
```

File clones of clones can also be created, as shown in the following example:

```
mmclone snap file1 snap1
mmclone copy snap1 file2
echo hello >> file2
mmclone snap file2 snap2
mmclone copy snap2 file3
```

The `echo` command updates the last block of file clone `file2`. When `file2` is snapped to `snap2`, the `mmclone snap` operation is performed as described previously. When a block in `file3` is read, the clone parent inode is found first. For the case of the last block, with the `hello` text, the disk address will be found in `snap2`. However, for other blocks, the disk address will be found in `snap1`.

For complete usage information, see the topic *mmclone command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Listing file clones

Use the `mmclone` command to display status for specified files.

The `show` keyword of the `mmclone` command provides a report to determine the current status of one or more files. When a file is a clone, the report will show the parent inode number. When a file was cloned from a file in a snapshot, `mmclone show` displays the snapshot and fileset information.

Consider the following scenario:

1. The `ls` command is issued to show all `.img` files in the current directory:

```
ls -ils *.img
```

The system displays output similar to the following:

```
148485 5752576 -rw-r--r-- 1 root root 21474836480 Jan 9 16:19 test01.img
```

2. A file clone is then created with the following commands:

```
mmclone snap test01.img base.img
mmclone copy base.img test02.img
```

3. After the file clone is created, the `mmclone show` command is issued to show information about all `.img` files in the current directory:

```
mmclone show *.img
```

The system displays output similar to the following:

Parent	Depth	Parent inode	File name
yes	0		base.img
no	1	148488	test01.img
no	1	148488	test02.img

4. A subsequent `ls` command would display output similar to the following:

```
# ls -ils *.img
148488 5752576 -rw-r--r-- 3 root root 21474836480 Jan 9 16:25 base.img
148485      0 -rw-r--r-- 1 root root 21474836480 Jan 9 16:19 test01.img
148480      0 -rw-r--r-- 1 root root 21474836480 Jan 9 16:25 test02.img
```

For complete usage information, see the topic *mmclone command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Deleting file clones

There is no explicit GPFS command available for deleting file clones. File clones can be deleted using a regular delete (`rm`) command. Clone parent files cannot be deleted until all file clone copies of the parent have been deleted and all open file handles to them have been closed.

Note: There is a brief period of time, immediately following the deletion of the file clone copies, when deletion of the parent can fail because the clone copy deletions are still running in the background.

Splitting file clones from clone parents

Use the `mmclone` command to split a file clone from a clone parent.

File clones can be split from their clone parents in one of two ways:

- Using the `mmclone redirect` command to split the file clone from the immediate clone parent only. The clone child remains a file clone, but the clone parent can be deleted.
- Using the `mmclone split` command to split the file clone from all clone parents. This converts the former clone child to a regular file. The clone parent does not change.

For complete usage information, see the topic *mmclone command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

File clones and disk space management

File clones have several considerations that are related to disk space management.

- Replication and storage pools

Each file clone has its own inode, so attributes and permissions for each clone can be set independently. For example, timestamps (atime, mtime, and ctime) are maintained separately for each clone. Clone parent attributes must be changed separately. If different clones have different values for replication or storage pool, it is not possible for every one of the clones to have all data blocks readable through that clone to be replicated and placed consistent with its replication and pool settings. Thus, changes to replication and storage pool only apply to blocks added to the clone and leave the clone parent unchanged.

- Clone ownership, block counts, and quotas

Creating a clone parent requires read access to the file being cloned. The person creating the clone parent does not have to be the owner of the original file, but will become the owner of the new clone parent. The block count and disk quota of the original file will be transferred to the new clone parent inode and then set to zero in the original file. Any blocks allocated by copy-on-write of a clone file will be added to the block count in the clone inode, with quota charged against the owner of the file. If even a single byte of data in a clone child is changed, the entire block will be copied from the parent.

- Clones and DMAPI

Clone parent files and clone copy files will be preserved across migrations and recalls to and from tape.

File clones and snapshots

When a snapshot is created and a file clone is subsequently updated, the previous state of the file clone will be saved in the snapshot.

When reading a file clone in the snapshot, the system will distinguish between the states of the clone:

- The data block has not been modified since the snapshot was taken, so look for the data in the same file in the next most recent snapshot or active file system.
- The file clone was updated, which indicates that the corresponding data block should be found in the clone parent. The system will look for the data in the clone parent within the same snapshot.

When a snapshot has file clones, those file clones should be deleted or split from their clone parents prior to deleting the snapshot. For more information, see “[Deleting file clones](#)” on page 615 and “[Splitting file clones from clone parents](#)” on page 615. A policy file can be created to help determine if a snapshot has file clones. For more information, see “[File clones and policy files](#)” on page 616.

File clones and policy files

Policy files can be created to examine clone attributes.

The following clone attributes can be examined in a policy file:

- The depth of the clone tree.
- If file is an immutable clone parent.
- The filesset ID of the clone parent.
- The inode number of the clone parent for the file.
- If the clone parent is in a snapshot.
- The snapshot ID of the clone parent.

See “[File attributes in SQL expressions](#)” on page 541 for more information about the clone attributes available for policy files.

The following example shows a policy file that can be created for displaying clone attributes for all files:

```
RULE EXTERNAL LIST 'x' EXEC ''  
RULE 'nonClone' LIST 'x' SHOW('nonclone') WHERE Clone_Parent_Inode IS NULL  
RULE 'normalClone' LIST 'x' SHOW(  
    'inum' || varchar(Clone_Parent_Inode) ||  
    'par' || varchar(Clone_Is_Parent) ||  
    'psn' || varchar(Clone_Parent_Is_Snap) ||  
    'dep' || varchar(Clone_Depth))  
    WHERE Clone_Parent_Inode IS NOT NULL AND Clone_Parent_Is_Snap == 0  
RULE 'snapClone' LIST 'x' SHOW(  
    'inum' || varchar(Clone_Parent_Inode) ||  
    'par' || varchar(Clone_Is_Parent) ||  
    'psn' || varchar(Clone_Parent_Is_Snap) ||  
    'dep' || varchar(Clone_Depth) ||  
    'Fid' || varchar(Clone_Parent_Filesset_Id) ||  
    'snap' || varchar(Clone_Parent_Snap_Id))  
    WHERE Clone_Parent_Inode IS NOT NULL AND Clone_Parent_Is_Snap != 0
```

If this policy file was called pol.file, the following command would display the clone attributes:

```
mmapplypolicy fs0 -P pol.file -I defer -f pol -L 0
```

Chapter 42. Scale Out Backup and Restore (SOBAR)

Scale Out Backup and Restore (SOBAR) is a specialized mechanism for data protection against disaster only for GPFS file systems that are managed by IBM Storage Protect for Space Management.

Note: Available on all IBM Storage Scale editions.

To protect a file system against disaster the following steps must be taken to ensure all data is safely stored in a second location:

1. Record the file system configuration with the **mmbackupconfig** command.
2. Ensure all file data is *pre-migrated* (see “[Pre-migrating files with external storage pools](#)” on page 575 for more information).
3. Perform a metadata image backup with the **mmimgbackup** command.

The **mmbackupconfig** command must be run prior to running the **mmimgbackup** command. No changes to file system configuration, filesets, quotas, or other settings should be done between running the **mmbackupconfig** command and the **mmimgbackup** command. To recover from a disaster, the **mmrestoreconfig** command must be run prior to running the **mmimgrestore** command. The file system being restored must have the same inode size and metadata block size as the file system that was backed up. Use the **mmrestoreconfig -F QueryResultFile** option to create the QueryResultFile. Use the example of the **mmcrlfs** command within the QueryResultFile to recreate your file system. After restoring the image data and adjusting quota settings, the file system can be mounted read-write, and the HSM system re-enabled to permit file data recall. Users may be permitted to access the file system, and/or the system administrator can manually recall file data with the IBM Storage Protect for Space Management command **dsmrecall**.

These commands cannot be run from a Windows node.

Backup procedure with SOBAR

This section provides a detailed example of the backup procedure that is used with SOBAR.

Throughout these procedures, the sample file system that is used is called `smallfs`. Where appropriate, replace this value with your file system name.

1. Back up the cluster configuration information.

The cluster configuration must be backed up by the administrator. The minimum cluster configuration information that is needed is: IP addresses, node names, roles, quorum and server roles, cluster-wide configuration settings from `mmchconfig`, cluster manager node roles, remote shell configuration, mutual ssh and rsh authentication setup, and the cluster UID. Complete configuration information can be found in the `mmsdrfs` file and CCR.

2. Preserve disk configuration information.

Disk configuration must also be preserved to recover a file system. The basic disk configuration information needed, for a backup that is intended for disaster recovery, is the number of disk volumes that were previously available and the sizes of those volumes. To recover from a complete file system loss, at least as much disk space as was previously available is needed for restoration. It is feasible to restore the image of a file system on to replacement disks only when the disk volumes available are of similar sizes. This enables to restore data to the new disks. At a minimum, the following disk configuration information is needed:

- Disk device names
- Disk device sizes
- The number of disk volumes
- NSD server configuration
- Disk RAID configurations

- Failure group designations
 - The `mmsdifs` file contents
3. Back up the GPFS file system configuration information.

In addition to the disks, the file system that is built on those volumes has configuration information that can be captured by using the `mmbackupconfig` command. This information includes block size, replication factors, number and size of disks, storage pool layout, file sets and junction points, policy rules, quota information, and a number of other file system attributes. The file system configuration information can be backed up into a single file by using a command similar to the following example:

```
mmbackupconfig smallfs -o /tmp/smallfs.bkpcfg.out925
```

Ensure to copy the temporary file that is created by the preceding command to a secure location so that it can be retrieved and used during a disaster recovery.

4. Pre-migrate all newer file data into secondary storage.

File contents in a space-managed GPFS reside in secondary storage that is managed by IBM Storage Protect. If IBM Storage Protect, disk and tape pools typically hold the offline images of migrated files. IBM Storage Protect can also be used to pre-migrate all newer file data into secondary storage, so that all files have either a migrated or pre-migrated status (XATTR) recorded, and their current contents are copied or updated into the secondary storage. The IBM Storage Protect command `dsimmigrate` can be used as follows:

```
dsimmigrate -Premigrate -Recursive /smallfs
```

Optionally check the status of the files that were pre-migrated with the previous command, use the following command:

```
dsmls /smallfs/*
```

5. Create a global snapshot of the live file system, to provide a quiescent image for image backup, by using a command similar to the following:

```
mmcrlsnapshot smallfs smallfssnap
```

6. Choose a staging area in which to save the GPFS metadata image files.

The image backup process stores each piece of the partial file system image backup in its own file in the shared work directory that is typically used by policy runs. These files can become large depending on the number of files in the file system. Also, because the file system that is holding this shared directory must be accessible to every node that is participating in the parallel backup task, it might also be a GPFS file system. It is imperative that the staging directories chosen are accessible to both the `tsapolicy` archiver process and the IBM Storage Protect Backup-Archive client. This staging directory is specified with the `-g` option of the `mmimgbackup` command.

7. Back up the file system image.

The following command backs up an image of the GPFS metadata from the file system by using a parallel policy that is run with the default IBM Storage Protect backup client to back up the file system metadata image:

```
mmimgbackup smallfs -S smallfssnap -g /u/user/backup -N aixnodes
```

The metadata of the file system, the directories, inodes, attributes, symlinks, and so on, are all captured in parallel by using the archive module extension feature of the `mmapplypolicy` command. After completing the parallel execution of the policy-driven archiving process, a collection of image files in this format will remain. These image files are gathered by the `mmimgbackup` command and archived to IBM Storage Protect automatically.

If you are using the `-N nodes` option, it is a good idea to use the same operating system when running `mmimgbackup`. Also, the directory that was created with the `-g GlobalWorkDirectory` option to store the image files must exist and must be accessible from all the nodes that are specified.

8. After the image backup is complete, delete the snapshot that is used for backup with the following command:

```
mmdeletesnapshot smallfs smallfssnap
```

Related concepts

Restore procedure with SOBAR

This section provides a detailed example of the restore procedure used with SOBAR.

Restore procedure with SOBAR

This section provides a detailed example of the restore procedure used with SOBAR.

In order to restore a file system, the configuration data stored from a previous run of `mmbackupconfig` and the image files produced from `mmimgbackup` must be accessible.

Throughout these procedures, the sample file system used is called `smallfs`. Where appropriate, replace this value with your file system name.

1. Restore the metadata image files from `mmimgbackup` and the backup configuration data from `mmbackupconfig` with a `dsmc` command similar to the following:

```
dsmc restore -subdir=yes /u/user/backup/8516/
```

2. Retrieve the base file system configuration information.

Use the `mmrestoreconfig` command to generate a configuration file, which contains the details of the former file system:

```
mmrestoreconfig Device -i InputFile -F QueryResultFile
```

3. Recreate NSDs if they are missing.

Using the output file generated in the previous step as a guide, the administrator might need to recreate NSD devices for use with the restored file system. In the output file, the NSD configuration section contains the NSD information; for example:

```
##### NSD configuration #####
## Disk descriptor format for the mmcrlnsd command.
## Please edit the disk and desired name fields to match
## your current hardware settings.
##
## The user then can uncomment the descriptor lines and
## use this file as input to the -F option.
#
# %nsd:
#   device=DiskName
#   nsd=nsd8
#   usage=dataAndMetadata
#   failureGroup=-1
#   pool=system
#
```

If changes are needed, edit the file in a text editor and follow the included instructions to use it as input to the `mmcrlnsd` command, then issue the following command:

```
mmcrlnsd -F StanzaFile
```

4. Recreate the base file system.

The administrator must recreate the initial file system. The output query file specified in the previous commands can be used as a guide. The following example shows the section of this file that is needed when recreating the file system:

```
##### File system configuration #####
## The user can use the predefined options/option values
## when recreating the file system. The option values
## represent values from the backed up file system.
```

```
#  
# mmcrlfs FS_NAME NSD_DISKS -j cluster -k posix -Q yes -L 4194304 --disable-fastea  
-T /fs2 -A no --inode-limit 278016#  
#  
# When preparing the file system for image restore, quota  
# enforcement must be disabled at file system creation time.  
# If this is not done, the image restore process will fail.
```

Do one of the following to recreate the file system:

- Edit the output file. Uncomment the `mmcrlfs` command, and specify the appropriate file system name and NSD disk(s). Remove the `-Q` option to ensure quotas are not enabled. Save the changes and run the file as a shell script:

```
sh OutputFile
```

- From the command line, issue an `mmcrlfs` command similar to the one in the output file, but specify the appropriate file system name and NSD disk(s). Do not specify the `-Q` option to ensure quotas are not enabled. The inode size and metadata block size must be the same in the file system in order to be restored as is in the original file system.

5. Restore essential file system configuration.

Using the `mmrestoreconfig` command, the essential file system configuration can be restored to the file system that was just created in the previous step. Quota is disabled in this step because the quota system must remain inactive until after the file system image has been restored. Filesets will also be restored and linked, if necessary, using a method specific for image restore. The `--image-restore` option should be used to restore the configuration data in the proper format for SOBAR; for example:

```
mmrestoreconfig smallfs -i /tmp/smallfs.bkpcfg.out925 --image-restore
```

6. Mount the file system in read-only mode for image restore with the following command:

```
mmmount smallfs -o ro
```

7. Perform the image restore; for example:

```
mmimgrestore smallfs /u/user/backup/8516/mmPolicy.8551.D4D85229
```

8. To optionally display the restored file system structure, use the following command:

```
ls -l /smallfs/*
```

The system displays information similar to the following:

```
-rw-r--r-- 1 root root 1024 Sep 25 11:34 /smallfs/1Kfile.1  
-rw-r--r-- 1 root root 1024 Sep 25 11:34 /smallfs/1Kfile.2  
-rwxr--r-- 1 root root 238 Sep 25 11:34 /smallfs/generateChecksums*
```

9. Unmount the file system with the following command:

```
mmumount smallfs
```

10. Restore quota configuration.

If any quota enforcement was used in the prior file system, it can be restored now using the `mmrestoreconfig` command. This step will not enable quotas if they were not in use at the time of the configuration backup. To restore the quota configuration, issue a command similar to the following:

```
mmrestoreconfig smallfs -i /tmp/smallfs.bkpcfg.out925 -Q only
```

11. Mount the file system in read-write mode with the following command:

```
mmmount smallfs
```

12. Delete the unusable IBM Storage Protect directory.

The .SpaceMan directory contains file stubs from the former space management control information. This directory must be deleted prior to restarting IBM Storage Protect. Use the following command:

```
rm -rf /smallfs/.SpaceMan
```

13. To optionally restart IBM Storage Protect, use the following command:

```
dsmmigfs restart
```

14. Resume IBM Storage Protect on the newly reconstructed file system, to resume managing disk space and to permit recall of files, with the following command:

```
dsmmigfs add /smallfs
```

15. To optionally display the managed file system from IBM Storage Protect, use the following command:

```
dsmls /smallfs/*
```

All files are currently in the migrate state.

16. To optionally begin recalling files by forcing a specific recall, use the following command:

```
dsmrecall -Recursive /smallfs/*
```

Related concepts

[Backup procedure with SOBAR](#)

This section provides a detailed example of the backup procedure that is used with SOBAR.

Chapter 43. Data Mirroring and Replication

The ability to detect and quickly recover from a massive hardware failure is of paramount importance to businesses that make use of real-time data processing systems.

GPFS provides a number of features that facilitate the implementation of highly-available GPFS environments capable of withstanding catastrophic hardware failures. By maintaining a replica of the file system's data at a geographically-separate location, the system sustains its processing using the secondary replica of the file system in the event of a total failure in the primary environment.

On a high level, a disaster-resilient GPFS cluster is made up of two or three distinct geographically separate hardware sites operating in a coordinated fashion. Two of the sites consist of GPFS nodes and storage resources holding a complete replica of the file system. If a third site is active, it consists of a single node and a single disk used as a tiebreaker for GPFS quorum. In the event of a catastrophic hardware failure that disables the operation of an entire site, and assuming the tiebreaker site remains operational, file system services fail over to the remaining subset of the cluster and continue serving the data using the replica of the file system that survived the disaster. However, if the tiebreaker fails during the disaster, the remaining number of nodes and disks is insufficient to satisfy the quorum rules and the surviving site loses access to the GPFS file system. A manual procedure is needed to instruct GPFS to disregard the existing quorum assignments and continue operating with whatever resources are available.

The secondary replica is maintained by one of several methods:

- Synchronous mirroring utilizing GPFS replication.

The data and metadata replication features of GPFS are used to implement synchronous mirroring between a pair of geographically-separate sites. The use of logical replication-based mirroring offers a generic solution that relies on no specific support from the disk subsystem beyond the basic ability to read and write data blocks.

- Synchronous mirroring utilizing storage-based replication.

Hardware replication creates persistent mirroring relationship between pairs of Logical Units (LUNs) on two subsystems connected over SAN or LAN links. All updates performed on the set of primary, source, or LUNs appear in the same order on the secondary, target, or disks in the target subsystem. Hardware replication provides for an exact bitwise replica of the content of the source as seen at the time of the failure on the target if the source volume fails. A range of technologies can be used to provide synchronous replication such as Metro Mirror on DS8000® or Storwize® or Synchronous Remote Mirroring on XIV®.

- Asynchronous mirroring utilizing GPFS-based replication.

Asynchronous replication functionality provides a similar crash consistent copy of data as synchronous replication but in normal operation the secondary copy of data will lag behind the primary by some period of time. For more information, see the topic *AFM-based Asynchronous Disaster Recovery (AFM DR)* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

- Asynchronous mirroring utilizing storage-based replication.

Asynchronous replication functionality provides a similar crash consistent copy of data as synchronous replication but in normal operation the secondary copy of data will lag behind the primary by some time. A range of technologies can be used to provide asynchronous replication such as Global Mirror on DS8000 or Storwize or Asynchronous Remote Mirroring on XIV.

- Point in time copy using storage-based functionality.

Periodic point-in-time copies of the file system are taken using the functionality such as FlashCopy® on the DS8000 or Storwize or Snapshot on XIV. This copy could be used as a source of a complete file system consistent backup to be taken to a remote site or could be used in conjunction with other replication capabilities to use for isolated testing of disaster recovery procedures.

The primary advantage of both synchronous mirroring methods is the minimization of the risk of permanent data loss. Both methods provide two consistent, up-to-date replicas of the file system,

each available for recovery if the other one fails. However, inherent to all solutions that synchronously mirror data over a wide area network link is the latency penalty that is induced by the replicated write I/Os. This makes both synchronous mirroring methods prohibitively inefficient for certain types of performance-oriented applications where there is a longer distance between sites. The asynchronous method effectively eliminates this penalty but in a situation where the primary site is lost, there might be updates that have not yet been transferred to the secondary site. Asynchronous replication will still provide a crash consistent and restartable copy of the primary data.

General considerations for using storage replication with GPFS

This topic describes the general considerations that need to be followed for using storage replication with IBM Storage Scale.

Different storage-level replication capabilities are available on both IBM and non-IBM storage systems. IBM provides storage-level replication functionality on the following platforms:

- The DS8000 provides synchronous replication with Metro Mirror and asynchronous replication with Global Mirror. Three and four site replication topologies are also possible by combining these functions. For more information, see [IBM DS8000 Documentation](#).
- The Storwize family of storage systems also provides a synchronous replication capability with Metro Mirror and has two versions of asynchronous replication called Global Mirror and Global Mirror with Change Volumes. Point in Time copy functionality is provided by FlashCopy. For more information, see [IBM Storwize V7000 Documentation](#).
- The XIV provides both synchronous and asynchronous Remote Replication and also provides point in time copy functionality referred to as Snapshot. For more information, see [IBM XIV Storage System documentation](#).

Note: In this document, synchronous replication is referred to as Metro Mirror, asynchronous replication is referred to as Global Mirror, and point in time copy functionality is referred to as FlashCopy.

Data integrity and the use of consistency groups

Disk based replication technologies provide a crash consistent copy of the replicated data. This means that the data that is not committed to the second site when a failure occurs may not be saved. However, a volume can be recovered to a recent point in time when all of the data was consistent.

A group of volumes that share a common recovery point is commonly called a consistency group. The storage controller ensures that after a failure, all of the volumes within a consistency group are recovered to the same point in time.

When using a storage-based replication with IBM Storage Scale, it is important to ensure that all the NSD's in a file system are contained within the same consistency group. This way the metadata NSD's are always in sync with the data NSD's after a failure.

Handling multiple versions of IBM Storage Scale data

This topic provides description on handling multiple versions of data in IBM Storage Scale.

The primary copy of a GPFS file system and a hardware replicated copy cannot coexist in the same GPFS cluster. A node can mount either the original copy of the file system or one of its replicas, but not both. This restriction has to do with the current implementation of the NSD-to-LUN mapping mechanism, which scans all locally attached disks, searching for a specific value (the NSD ID) at a particular location on disk. If both the original volume and a hardware replica are visible to a particular node, these disks would appear to GPFS as distinct devices with identical NSD IDs.

For this reason, users are asked to zone their SAN configurations such that at most one replica of any given GPFS disk is visible from any node. That is, the nodes in your production cluster should have access to the disks that make up the actual file system but should not see the disks that hold the replicated copies, whereas the backup server should see the replication targets but not the originals.

Alternatively, you can use the **nsddevices** user exit located in `/var/mmfs/etc/` to explicitly define the subset of the locally visible disks to be accessed during the NSD device scan on the local node.

The following procedure is used to define an **nsddevices** user exit file to instruct GPFS to use a specific disk **diskA1** rather than other copies of this device, which might also be available:

```
echo "echo diskA1 hdisk" > /var/mmfs/etc/nsddevices chmod 744 /var/mmfs/etc/nsddevices
```

Refer to the prolog of `/usr/lpp/mmfs/samples/nsddevices.samples` for detailed instructions on the usage of **nsddevices**.

Continuous Replication of IBM Storage Scale data

This topic provides a brief description on replication of IBM Storage Scale data.

Synchronous mirroring with GPFS replication

In a configuration utilizing GPFS replication, a single GPFS cluster is defined over three geographically-separate sites consisting of two production sites and a third tiebreaker site. One or more file systems are created, mounted, and accessed concurrently from the two active production sites.

The data and metadata replication features of GPFS are used to maintain a secondary copy of each file system block, relying on the concept of disk failure groups to control the physical placement of the individual copies:

1. Separate the set of available disk volumes into two failure groups. Define one failure group at each of the active production sites.
2. Create a replicated file system. Specify a replication factor of 2 for both data and metadata.

When allocating new file system blocks, GPFS always assigns replicas of the same block to distinct failure groups. This provides a sufficient level of redundancy allowing each site to continue operating independently should the other site fail.

GPFS enforces a node quorum rule to prevent multiple nodes from assuming the role of the file system manager in the event of a network partition. Thus, a majority of quorum nodes must remain active in order for the cluster to sustain normal file system usage. Furthermore, GPFS uses a quorum replication algorithm to maintain the content of the file system descriptor (one of the central elements of the GPFS metadata). When formatting the file system, GPFS assigns some number of disks (usually three) as the descriptor replica holders that are responsible for maintaining an up-to-date copy of the descriptor. Similar to the node quorum requirement, a majority of the replica holder disks must remain available at all times to sustain normal file system operations. This file system descriptor quorum is internally controlled by the GPFS daemon. However, when a disk has failed due to a disaster you must manually inform GPFS that the disk is no longer available and it should be excluded from use.

Considering these quorum constraints, it is suggested that a third site in the configuration fulfill the role of a tiebreaker for the node and the file system descriptor quorum decisions. The tiebreaker site consists of:

1. A single quorum node

As the function of this node is to serve as a tiebreaker in GPFS quorum decisions, it does not require normal file system access and SAN connectivity. To ignore disk access errors on the tiebreaker node, enable the **unmountOnDiskFail** configuration parameter through the `mmchconfig` command. When enabled, this parameter forces the tiebreaker node to treat the lack of disk connectivity as a local error, resulting in a failure to mount the file system, rather than reporting this condition to the file system manager as a disk failure.

2. A single network shared disk

The function of this disk is to provide an additional replica of the file system descriptor file needed to sustain quorum should a disaster cripple one of the other descriptor replica disks. Create a network shared disk over the tiebreaker node's internal disk defining:

- the local node as an NSD server

- the disk usage as **descOnly**

The **descOnly** option instructs GPFS to only store file system descriptor information on the disk.

This three-site configuration is resilient to a complete failure of any single hardware site. Should all disk volumes in one of the failure groups become unavailable, GPFS performs a transparent failover to the remaining set of disks and continues serving the data to the surviving subset of nodes with no administrative intervention. While nothing prevents you from placing the tiebreaker resources at one of the active sites, to minimize the risk of double-site failures it is suggested you install the tiebreakers at a third, geographically distinct location.

Important: Note the following good practices:

- In an environment that is running synchronous mirroring using GPFS replication:
 - Do not designate a tiebreaker node as a manager node.
 - Do not mount the file system on the tiebreaker node. To avoid unexpected mounts, you can create the following file with any content on the tiebreaker node:

```
/var/mmfs/etc/ignoreAnyMount.<file_system_name>
```

In the following example, the file system is `fs1`:

```
/var/mmfs/etc/ignoreAnyMount.fs1
```

For more information, see [Chapter 26, “Managing file systems,” on page 217](#).

Note: If you create an `ignoreAnyMount.<file_system_name>` file, you cannot manually mount the file system on the tiebreaker node.

If you do not follow these practices, an unexpected file system unmount can occur during site failures, because of the configuration of the tiebreaker node and the `unmountOnDiskFail` option.

- In a stretch cluster environment, designate at least one quorum node from each site as a manager node. During site outages, the quorum nodes can take over as manager nodes.

Note: There are no special networking requirements for this configuration. For example:

- You do not need to create different subnets.
- You do not need to have GPFS nodes in the same network across the two production sites.
- The production sites can be on different virtual LANs (VLANs).

Limitation:

- If the Object protocol is deployed on the cluster and the CES networks of two production sites cannot communicate with each other, you must change the Object Ring configuration to use the CES IP addresses of only one of the production sites. Follow the procedure that is described in the topic [“Configuration of object for isolated node and network groups” on page 420](#).
- Clustered watch folder is not supported in a stretch cluster environment.

The high-level organization of a replicated GPFS cluster for synchronous mirroring where all disks are directly attached to all nodes in the cluster is shown in [Figure 19 on page 627](#). An alternative to this design would be to have the data served through designated NSD servers.

With GPFS release 4.1.0, a new, more fault-tolerant configuration mechanism has been introduced as the successor for the server-based mechanisms. The server-based configuration mechanisms consist of two configuration servers specified as the primary and secondary cluster configuration server. The new configuration mechanism uses all specified quorum nodes in the cluster to hold the GPFS configuration and is called CCR (Clustered Configuration Repository). The CCR is used by default during cluster creation unless the CCR is explicitly disabled. The `mmlscluster` command reports the configuration mechanism in use in the cluster.

The following sections describe the differences regarding disaster recovery for the two configuration mechanisms.

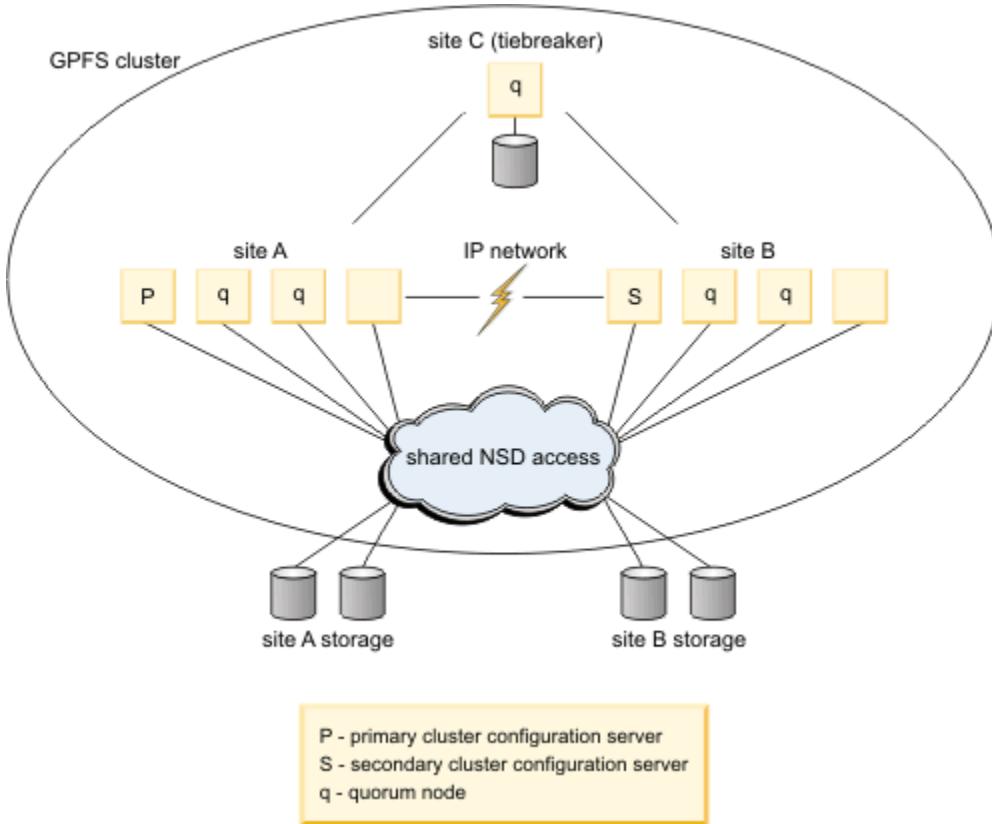


Figure 19. Synchronous mirroring utilizing GPFS replication

Setting up IBM Storage Scale synchronous replication

See an example of how to set up synchronous IBM Storage Scale replication to recover from a site failure.

Synchronous replication is described in the topic and in Figure 1 of that topic.

This example is based on the following configuration:

Site A

- Nodes: **nodeA001, nodeA002, nodeA003, nodeA004**
- Disk devices: **diskA1** and **diskA2**. These devices are SAN-attached and accessible from all the nodes at site **A** and site **B**.

Site B

- Nodes: **nodeB001, nodeB002, nodeB003, node B004**
- Disk devices: **diskB1** and **diskB2**. These devices are SAN-attached and accessible from all the nodes at site **A** and site **B**.

Site C

Note that site **C** contains only one node, which will be defined as a quorum node and a client node.

- Nodes: **nodeC**
- Disks: **diskC**. This disk is an NSD defined on an internal disk of **nodeC** and is directly accessible only from site **C**.

1. Create an IBM Storage Scale cluster with the **mmcrcluster** command and a node file.

a) Create a node file that is named **clusterNodes** with the following contents:

```
nodeA001:quorum-manager
nodeA002:quorum-manager
nodeA003:quorum-manager
nodeA004:client
```

```
nodeB001:quorum-manager  
nodeB002:quorum-manager  
nodeB003:quorum-manager  
nodeB004:client  
nodeC:quorum-client
```

- b) Issue the following command to create the cluster:

```
mmcrcluster -N clusterNodes
```

Note: The cluster is created with the Cluster Configuration Repository (CCR) enabled. This option is the default on IBM Storage Scale 4.1 or later.

2. Issue the following command to enable the **unmountOnDiskFile** attribute on **nodeC**:

```
mmchconfig unmountOnDiskFail=yes -N nodeC
```

Enabling this attribute prevents false disk errors in the SAN configuration from being reported to the file system manager.

Important: In a synchronous replication environment, the following rules are good practices:

- The following rules apply to **nodeC**, which is the only node on site **C** and is also a client node and a quorum node:
 - Do not designate the **nodeC** as a manager node.
 - Do not mount the file system on **nodeC**.

To avoid unexpected mounts, create the following empty file on **nodeC**:

```
/var/mmfs/etc/ignoreAnyMount.<file_system_name>
```

For example, if the file system is **fs0**, create the following empty file:

```
/var/mmfs/etc/ignoreAnyMount.fs0
```

Note: If you create an **ignoreAnyMount.<file_system_name>** file, you cannot manually mount the file system on **nodeC**.

If you do not follow these practices, an unexpected file system unmount can occur during site failures, because of the configuration of **nodeC** and the **unmountOnDiskFail** option.

- In the sites that do not contain a single quorum client node (sites **A** and **B** in this example), designate at least one quorum node from each site as a manager node. During a site outage, a quorum node can take over as a manager node.

3. Create a set of network shared disks (NSDs) for the cluster.

- a) Create the stanza file **clusterDisks** with the following NSD stanzas:

```
%nsd: device=/dev/diskA1  
servers=nodeA002,nodeA003  
usage=dataAndMetadata  
failureGroup=1  
%nsd: device=/dev/diskA2  
servers=nodeA003,nodeA002  
usage=dataAndMetadata  
failureGroup=1  
%nsd: device=/dev/diskB1  
servers=nodeB002,nodeB003  
usage=dataAndMetadata  
failureGroup=2  
%nsd: device=/dev/diskB2  
servers=nodeB003,nodeB002  
usage=dataAndMetadata  
failureGroup=2  
%nsd: device=/dev/diskC1  
servers=nodeC  
usage=descOnly  
failureGroup=3
```

Important: Note that the stanzas make the following failure group assignments:

- The disks at site **A** are assigned to failure group 1.
- The disks at site **B** are assigned to failure group 2.
- The disk that is local to **nodeC** is assigned to failure group 3.

b) Issue the following command to create the NSDs:

```
mmcrnsd -F clusterDisks
```

c) Issue the following command to verify that the network shared disks are created:

```
mmlsnsd -m
```

The command should display output like the following:

Disk name	NSD volume ID	Device	Node name	Remarks
gpfs1nsd	0972445B416BE502	/dev/diskA1	nodeA002	server node
gpfs1nsd	0972445B416BE502	/dev/diskA1	nodeA003	server node
gpfs2nsd	0972445B416BE509	/dev/diskA2	nodeA002	server node
gpfs2nsd	0972445B416BE509	/dev/diskA2	nodeA003	server node
gpfs3nsd	0972445F416BE4F8	/dev/diskB1	nodeB002	server node
gpfs3nsd	0972445F416BE4F8	/dev/diskB1	nodeB003	server node
gpfs4nsd	0972445F416BE4FE	/dev/diskB2	nodeB002	server node
gpfs4nsd	0972445F416BE4FE	/dev/diskB2	nodeB003	server node
gpfs5nsd	0972445D416BE504	/dev/diskC1	nodeC	server node

4. Issue the following command to start IBM Storage Scale on all the nodes of the cluster:

```
mmstartup -a
```

5. Create a file system **fs0** with default replication for metadata (-m 2) and data (-r 2).

Issue the following command:

```
mmcrifs /gpfs/fs0 fs0 -F clusterDisks -m 2 -r 2
```

6. Mount the file system **fs0** on all the cluster nodes at site **A** and site **B**.

7. This step is optional and for ease of use.

Issue the following three commands to create node classes for sites **A**, **B**, and **C**:

```
mmcrinodeclass gpfs.siteA -N  
prt001st001,prt002st001,prt003st001,prt004st001,nsd001st001,nsd002st001
```

```
mmcrinodeclass gpfs.siteB -N  
prt008st001,prt007st001,prt006st001,prt005st001,nsd004st001,nsd003st001
```

```
mmcrinodeclass gpfs.siteC -N nsd005st001
```

You can now use node class names with IBM Storage Scale commands ("mm" commands) to recover sites easily after a cluster failover and failback. For example, with the following command you can bring down all the nodes on site **B** with one parameter, rather than having to pass all the node names for site **B** into the command:

```
mmshutdown -N gpfs.siteB
```

For information on the recovery procedure, see [“Failback with temporary loss using the Clustered Configuration Repository \(CCR\) configuration mechanism” on page 632](#).

The cluster is configured with synchronous replication to recover from a site failure.

Steps to take after a disaster when using IBM Storage Scale replication

Utilizing GPFS replication allows for *failover* to the surviving site without disruption of service as long as both the remaining site and the tiebreaker site remain functional. It remains in this state until a decision is made to restore the operation of the affected site by executing the *failback* procedure. If the tiebreaker

site is also affected by the disaster and is no longer operational, GPFS quorum is broken and manual intervention is required to resume file system access.

Existing quorum designations must be relaxed in order to allow the surviving site to fulfill quorum requirements:

1. To relax node quorum, temporarily change the designation of each of the failed quorum nodes to non-quorum nodes. Issue the `mmchnode --nonquorum` command.
2. To relax file system descriptor quorum, temporarily eliminate the failed disks from the group of disks from which the GPFS daemon uses to write the file system descriptor file to. Issue the `mmfsctl exclude` command for each of the failed disks.

While the GPFS cluster is in a failover state, it is suggested that no changes to the GPFS configuration be made. If the server-based configuration mechanism is in use, changes to your GPFS configuration require both cluster configuration servers to be operational. If both servers are not operational, the sites would have distinct, and possibly inconsistent, copies of the GPFS **mmsdrfs** configuration data file. While the servers can be migrated to the surviving site, it is best to avoid this step if the disaster does not leave the affected site permanently disabled.

If it becomes absolutely necessary to modify the GPFS configuration while in failover mode, for example to relax quorum, you must ensure that all nodes at the affected site are powered down and left in a stable inactive state. They must remain in such state until the decision is made to execute the failback procedure. As a means of precaution, we suggest disabling the GPFS autoload option on all nodes to prevent GPFS from bringing itself up automatically on the affected nodes should they come up spontaneously at some point after a disaster.

Failover to the surviving site

Following a disaster, which failover process is implemented depends upon whether or not the tiebreaker site is affected.

Failover without the loss of tiebreaker site C

The proposed three-site configuration is resilient to a complete failure of any single hardware site. Should all disk volumes in one of the failure groups become unavailable, GPFS performs a transparent failover to the remaining set of disks and continues serving the data to the surviving subset of nodes with no administrative intervention.

Failover with the loss of tiebreaker site C with server-based configuration in use

If both site **A** and site **C** fail:

1. Shut the GPFS daemon down on the surviving nodes at site **B**, where the file **gpfs.siteB** lists all of the nodes at site **B**:

```
mmshutdown -N gpfs.siteB
```

2. If it is necessary to make changes to the configuration, migrate the primary cluster configuration server to a node at site **B**:

```
mmchcluster -p nodeB002
```

3. Relax node quorum by temporarily changing the designation of each of the failed quorum nodes to non-quorum nodes:

```
mmchnode --nonquorum -N nodeA001,nodeA002,nodeA003,nodeC
```

4. Relax file system descriptor quorum by informing the GPFS daemon to migrate the file system descriptor off of the failed disks:

```
mmfsctl fs0 exclude -d "gpfs1nsd;gpfs2nsd;gpfs5nsd"
```

5. Restart the GPFS daemon on the surviving nodes:

```
mmstartup -N gpfs.siteB
```

6. Mount the file system on the surviving nodes at site B.

Failover with the loss of tiebreaker site C with Clustered Configuration Repository (CCR) in use

If both site A and site C fail:

1. Shut the GPFS daemon down on the surviving nodes at site B , where the file gpfs.siteB lists all of the nodes at site B :

```
mmdsh -N gpfs.siteB /usr/lpp/mmfs/bin/mmshutdown
```

2. Changing (downgrading) the quorum assignments when half or more of the quorum nodes are no longer available at site B using the -- force option :

```
mmchnode --nonquorum -N nodeA001,nodeA002,nodeA003,nodeC --force
```

3. Relax file system descriptor quorum by informing the GPFS daemon to migrate the file system descriptor off of the failed disks:

```
mmfsctl fs0 exclude -d "gpfs1nsd;gpfs2nsd;gpfs5nsd"
```

4. Restart the GPFS daemon on the surviving nodes:

```
mmstartup -N gpfs.siteB
```

5. Mount the file system on the surviving nodes at site B.

Make no further changes to the quorum designations at site B until the failed sites are back online and the following failback procedure has been completed.

Do not shut down the current set of nodes on the surviving site B and restart operations on the failed sites A and C. This will result in a non-working cluster.

Fallback procedures

Which failback procedure you follow depends upon whether the nodes and disks at the affected site have been repaired or replaced.

If the disks have been repaired, you must also consider the state of the data on the failed disks:

- For nodes and disks that have been repaired and *you are certain* the data on the failed disks has not been changed, follow either:
 - *failback with temporary loss and no configuration changes*
 - *failback with temporary loss and configuration changes*
- If the nodes have been replaced and either the disks have been replaced or repaired, and *you are not certain* the data on the fail disks has not been changed, follow the procedure for *failback with permanent loss*.

Delayed failures: In certain failure cases the loss of data may not be immediately apparent. For example, consider this sequence of events:

1. Site **B** loses connectivity with sites **A** and **C**.
2. Site **B** then goes down due to loss of node quorum.
3. Sites **A** and **C** remain operational long enough to modify some of the data on disk but suffer a disastrous failure shortly afterwards.
4. Node and file system descriptor quorums are overridden to enable access at site **B**.

Now the two replicas of the file system are inconsistent and the only way to reconcile these copies during recovery is to:

1. Remove the damaged disks at sites **A** and **C**.
2. Either replace the disk and format a new NSD or simply reformat the existing disk if possible.
3. Add the disk back to the file system, performing a full resynchronization of the file system's data and metadata and restore the replica balance using the `mmrestripefs` command.

Fallback with temporary loss and no configuration changes

If the outage was of a temporary nature and your configuration has not been altered, it is a simple process to fail back to the original state.

After all affected nodes and disks have been repaired and you *are certain* the data on the failed disks has not been changed:

1. Start GPFS on the repaired nodes where the file `gpfs.sitesAC` lists all of the nodes at sites **A** and **C**:

```
mmstartup -N gpfs.sitesAC
```

2. Restart the affected disks. If more than one disk in the file system is down, they must all be started at the same time:

```
mmchdisk fs0 start -a
```

Related tasks

[Fallback with temporary loss using the Clustered Configuration Repository \(CCR\) configuration mechanism](#)

If the outage was of a temporary nature and your configuration has been altered, follow this procedure to failback to the original state, in case the Clustered Configuration Repository (CCR) configuration scheme is in use.

Related reference

[Fallback with permanent loss](#)

If an outage is of a permanent nature, follow steps to remove and replace the failed resources, and then resume the operation of GPFS across the cluster.

Fallback with temporary loss using the Clustered Configuration Repository (CCR) configuration mechanism

If the outage was of a temporary nature and your configuration has been altered, follow this procedure to failback to the original state, in case the Clustered Configuration Repository (CCR) configuration scheme is in use.

After all affected nodes and disks have been repaired and you are certain the data on the failed disks has not been changed, complete the following steps.

1. Shut down the GPFS daemon on the surviving nodes at site B, and on the former failed and now recovered sites A and C, where the file `gpfs.siteB` lists all of the nodes at site B and the file `gpfs.siteA` lists all of the nodes at site A and the tiebreaker node at site C:

```
mmshutdown -N gpfs.siteB  
mmshutdown -N gpfs.siteA  
mmshutdown -N nodeC
```

2. Restore original node quorum designation for the tiebreaker site C at site B and start GPFS on site C:

```
mmstartup -N gpfs.siteB  
mmchnode --quorum -N nodeC  
mmstartup -N nodeC
```

3. Restore original node quorum designation for site A at site B and start GPFS on site A:

```
mmchnode --quorum -N nodeA001,nodeA002,nodeA003  
mmstartup -N gpfs.siteA
```

4. Restore the file system descriptor quorum by informing the GPFS to include the repaired disks:

```
mmumount fs0 -a;mmfsctl fs0 include -d "gpfs1nsd;gpfs2nsd;gpfs5nsd"
```

5. Mount the file system on all nodes at sites A and B.

Note: Do not allow the failed sites A and C to come online at the same time or when site B is unavailable or not functional.

6. Bring the disks online and restripe the file system across all disks in the cluster to restore the initial replication properties:

```
mmchdisk fs0 start -a  
mmrestripefs fs0 -b
```

The `-r` flag can be used on the `mmrestripefs` command instead.

Related tasks

Failback with temporary loss and no configuration changes

If the outage was of a temporary nature and your configuration has not been altered, it is a simple process to fail back to the original state.

Related reference

Failback with permanent loss

If an outage is of a permanent nature, follow steps to remove and replace the failed resources, and then resume the operation of GPFS across the cluster.

Failback with permanent loss

If an outage is of a permanent nature, follow steps to remove and replace the failed resources, and then resume the operation of GPFS across the cluster.

1. Remove the failed resources from the GPFS configuration.
2. Replace the failed resources, then add the new resources into the configuration.
3. Resume the operation of GPFS across the entire cluster.

Assume that sites **A** and **C** have had permanent losses. To remove all references of the failed nodes and disks from the GPFS configuration and replace them:

Procedure when Clustered Configuration Repository (CCR) is in use

1. To remove the failed resources from the GPFS configuration:

- a. Delete the failed disks from the GPFS configuration:

```
mmdeldisk fs0 "gpfs1nsd;gpfs2nsd;gpfs5nsd"
```

```
mmdelnsd "gpfs1nsd;gpfs2nsd;gpfs5nsd"
```

- b. Delete the failed nodes from the GPFS configuration:

```
mmdelnod -N nodeA001,nodeA002,nodeA003,nodeA004,nodeC
```

2. If there are new resources to add to the configuration:

- a. Add the new nodes at sites **A** and **C** to the cluster where the file **gpfs.sitesAC** lists the new nodes:

```
mmaddnode -N gpfs.sitesAC
```

- b. Restore original quorum node assignments at site B:

```
mmchnode --quorum -N nodeA001,nodeA002,nodeA003,nodeC
```

- c. Start GPFS on the new nodes

```
mmstartup -N gpfs.sitesAC
```

- d. Prepare the new disks for use in the cluster, create the NSDs using the original disk descriptors for site **A** contained in the file **clusterDisksAC**:

```
%nsd: device=/dev/diskA1  
servers=nodeA002,nodeA003  
usage=dataAndMetadata  
failureGroup=1  
  
%nsd: device=/dev/diskA2  
servers=nodeA003,nodeA002  
usage=dataAndMetadata  
failureGroup=1  
  
%nsd: device=/dev/diskC1  
servers=nodeC  
usage=descOnly  
  
failureGroup=3mmcrnsd -F clusterDisksAC
```

- e. Add the new NSDs to the file system specifying the **-r** option to rebalance the data on all disks:

```
mmadddisk fs0 -F clusterDisksAC -r
```

Related tasks

[Failback with temporary loss and no configuration changes](#)

If the outage was of a temporary nature and your configuration has not been altered, it is a simple process to fail back to the original state.

[Failback with temporary loss using the Clustered Configuration Repository \(CCR\) configuration mechanism](#)

If the outage was of a temporary nature and your configuration has been altered, follow this procedure to failback to the original state, in case the Clustered Configuration Repository (CCR) configuration scheme is in use.

Synchronous mirroring utilizing storage based replication

This topic describes synchronous mirroring utilizing storage-based replication.

Synchronous replication in the storage layer continuously updates a secondary (target) copy of a disk volume to match changes made to a primary (source) volume. A pair of volumes are configured in a replication relationship, during which all write operations performed on the source are synchronously mirrored to the target device.

The synchronous replication protocol guarantees that the secondary copy is constantly up-to-date by ensuring that the primary copy is written only if the primary storage subsystem received an acknowledgment that the secondary copy has been written. The paired volumes typically reside on two distinct and geographically separated storage systems communicating over a SAN or LAN link.

Most synchronous replication solutions provide a capability to perform an incremental resynchronization of data when switching between primary and secondary storage systems. After the failure of the primary volume (or the failure of the entire storage subsystem or site), users perform a failover, which suspends the relationship between the given pair of volumes and turns the target volume into a primary. When a volume enters the suspended state, a modification bitmap is established to keep track of the write operations performed on that volume to allow for an efficient resynchronization.

Once the operation of the original primary volume has been restored, a failback is executed to resynchronize the content of the two volumes. The original source volume is switched to the target mode, after which all modified data tracks (those recorded in the modification bitmap) are copied from the original target disk. The volume pair can then be suspended again and a similar process performed to reverse the volumes' roles, thus bringing the pair into its initial state.

A GPFS cluster using hardware-based replication can be established in two manners:

- A single GPFS cluster encompassing two sites and an optional tiebreaker site
- Two distinct GPFS clusters

An active-active IBM Storage Scale cluster

In an active-active cluster, a single GPFS cluster contains two active sites and an optional tiebreaker site.

The high-level organization of an active/active GPFS cluster using hardware replication is illustrated in Figure 20 on page 635. A single GPFS cluster is created over three sites. The data is mirrored between two active sites with a cluster configuration server residing at each site and a tiebreaker quorum node installed at the third location. The presence of an optional tiebreaker node allows the surviving site to satisfy the node quorum requirement with no additional intervention. Without the tiebreaker, the failover procedure requires an additional administrative command to relax node quorum and allow the remaining site to function independently. Furthermore, the nodes at the recovery site have direct disk paths to the primary site's storage.

The GPFS configuration resides either on the two configuration server (primary and secondary), when the cluster has been created with the Clustered Configuration Repository (CCR) disable option (**mmcrcluster**), or on each quorum node, when the cluster has Clustered Configuration Repository (CCR) enabled, or on the primary/secondary, when the Clustered Configuration Repository (CCR) is disabled.

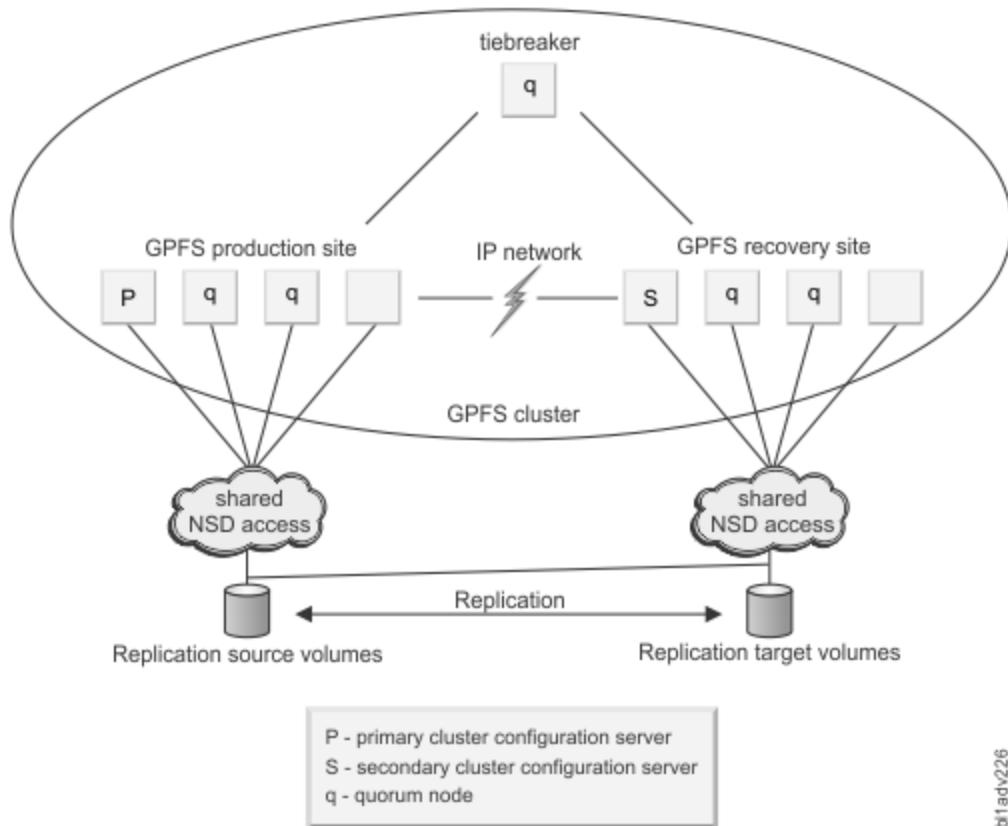


Figure 20. A synchronous active-active replication-based mirrored GPFS configuration with a tiebreaker site

Setting up an active-active GPFS configuration

This example demonstrates how to configure an active-active GPFS cluster.

To establish an active-active GPFS cluster using hardware replication with a tiebreaker site as shown in Figure 20 on page 635, consider the configuration:

Site A (production site)

Consists of:

- Nodes – **nodeA001, nodeA002, nodeA003, nodeA004**
- Storage subsystems – **A**
- Disk volumes – **diskA** on storage system **A**

diskA is SAN-attached and accessible from sites **A** and **B**

Site B (recovery site)

Consists of:

- Nodes – **nodeB001, nodeB002, nodeB003, nodeB004**
- Storage subsystems – **B**
- Disk volumes – **diskB** on storage system **B**

diskB is SAN-attached and accessible from site **B** only

Site C (tiebreaker)

Consists of:

- Nodes – **nodeC**

diskC is an NSD defined over the internal disk of the node **nodeC** and is directly accessible only from site **C**

1. Establish a hardware replication connectivity between the storage systems and then establish the synchronous replication volume pair between the source and target using the copy entire volume option. In this case, it would be **diskA–diskB**.
2. In order to protect the order of dependent writes that span multiple disk volumes, multiple storage systems, or both, the consistency group functionality of the storage system should be used with all GPFS devices in the same consistency group.
3. Create a GPFS cluster defining the primary cluster configuration server as nodes **nodeA001** at site **A**, the secondary cluster configuration server as **nodeB001** at site **B**, an equal number of quorum nodes at each site, including the tiebreaker node at site **C**, **nodeC**. To prevent the tiebreaker node from assuming the role of file system manager, define it as **client**. Define all other quorum nodes as **manager**. List the nodes in the cluster in the file **NodeDescFile**. The **NodeDescFile** file contains the node descriptors:

```
nodeA001:quorum-manager  
nodeA002:quorum-manager  
nodeA003:quorum-manager  
nodeA004:client  
nodeB001:quorum-manager  
nodeB002:quorum-manager  
nodeB003:quorum-manager  
nodeB004:client  
nodeC:quorum-client
```

Issue this command:

```
mmcircluster -N NodeDescFile -p nodeA001 -s nodeB001
```

4. On the tiebreaker node, issue the **mmchconfig** command to set the **unmountOnDiskFail** attribute to **yes**:

```
mmchconfig unmountOnDiskFail=yes -N nodeC
```

This action prevents false disk errors in the SAN configuration from being reported to the file system manager.

5. Enable the **unmountOnDiskFail** option on the tiebreaker node preventing false disk errors in the SAN configuration from being reported to the file system manager by issuing the **mmchconfig** command:

```
mmchconfig unmountOnDiskFail=yes -N nodeC
```

6. Create an NSD over **diskA**. The disk descriptor contained in the file **DiskDescFile** is:

```
/dev/diskA:nodeA001:nodeA002:dataAndMetadata:1
```

Issue this command:

```
mmcrnsd -F DiskDescFileP
```

7. Start the GPFS daemon on all nodes:

```
mmstartup -a
```

8. Create a GPFS file system and mount it on all nodes at sites **A** and **B**.

```
mmcrfs /gpfs/fs0 fs0 -F DiskDescFile
```

Failover to the recovery site and subsequent failback for an active/active configuration

For an active/active storage replication based cluster, complete these steps to restore access to the file system through site **B** after site **A** has experienced a disastrous failure.

Procedure when the server-based configuration scheme is in use

1. Stop the GPFS daemon on the surviving nodes as site **B** where the file **gpfs.siteB** lists all of the nodes at site **B**:

```
mmdsh -N gpfs.siteB /usr/lpp/mmfs/bin/mmshutdown
```

2. Perform the appropriate commands to make the secondary replication devices available and change their status from being secondary devices to suspended primary devices.
3. If you needed to relax node quorum or make configuration changes, migrate the primary cluster configuration server to site **B**, issue this command:

```
mmchcluster -p nodeB001
```

4. If site **C**, the tiebreaker, failed along with site **A**, existing node quorum designations must be relaxed in order to allow the surviving site to fulfill quorum requirements. To relax node quorum, temporarily change the designation of each of the failed quorum nodes to non-quorum nodes:

```
mmchnode --nonquorum -N nodeA001,nodeA002,nodeA003,nodeC
```

5. Ensure the source volumes are *not* accessible to the recovery site:

- Disconnect the cable
- Define the nsddevices user exit file to exclude the source volumes

6. Restart the GPFS daemon on all surviving nodes:

```
mmstartup -N gpfs.siteB
```

Procedure when the Clustered Configuration Repository (CCR) scheme is in use

For an active-active PPRC-based cluster, follow these steps to restore access to the file system through site **B** after site **A** has experienced a disastrous failure:

1. Stop the GPFS daemon on the surviving nodes as site **B** where the file **gpfs.siteB** lists all of the nodes at site **B**:

```
mmshutdown -N gpfs.siteB
```

2. Perform the appropriate commands to make the secondary replication devices available and change their status from being secondary devices to suspended primary devices.
3. If site **C**, the tiebreaker, failed along with site **A**, existing node quorum designations must be relaxed in order to allow the surviving site to fulfill quorum requirements. To relax node quorum, temporarily change the designation of each of the failed quorum nodes to nonquorum nodes using the **-- force** option:

```
mmchnode --nonquorum -N nodeA001,nodeA002,nodeA003,nodeC --force
```

4. Ensure that the source volumes are not accessible to the recovery site:

- Disconnect the cable.
- Define the **nsddevices** user exit file to exclude the source volumes.

5. Restart the GPFS daemon on all surviving nodes:

```
mmstartup -N gpfs.siteB
```

Note:

- Make no further changes to the quorum designations at site B until the failed sites are back online and the following failback procedure has been completed.
- Do not shut down the current set of nodes on the surviving site B and restart operations on the failed sites A and C. This will result in a non-working cluster.

Failback procedure

After the operation of site **A** has been restored, the failback procedure is completed to restore the access to the file system from that location. The following procedure is the same for both configuration schemes (server-based and Clustered Configuration Repository (CCR)). The failback operation is a two-step process:

1. For each of the paired volumes, resynchronize the pairs in the reserve direction with the recovery LUN **diskB** acting as the sources for the production LUN **diskA**. An incremental resynchronization is performed, which identifies the mismatching disk tracks, whose content is then copied from the recovery LUN to the production LUN. Once the data has been copied and the replication is running in the reverse direction this configuration can be maintained until a time is chosen to switch back to site **A**.
2. Shut GPFS down at site **B** and reverse the disk roles (the original primary disk becomes the primary again), bringing the replication pair to its initial state.
 - a. Stop the GPFS daemon on all nodes.
 - b. Perform the appropriate actions to switch the replication direction so that **diskA** is now the source and **diskB** is the target.
 - c. If during failover you migrated the primary cluster configuration server to a node in site **B**:
 - i) Migrate the primary cluster configuration server back to site **A**:

```
mmchcluster -p nodeA001
```

ii) Restore the initial quorum assignments:

```
mmchnode --quorum -N nodeA001,nodeA002,nodeA003,nodeC
```

iii) Ensure that all nodes have the latest copy of the **mmsdrfs** file:

```
mmchcluster -p LATEST
```

d. Ensure the source volumes *are* accessible to the recovery site:

- Reconnect the cable
- Edit the nsddevices user exit file to *include* the source volumes

e. Start the GPFS daemon on all nodes:

```
mmstartup -a
```

f. Mount the file system on all the nodes at sites **A** and **B**.

An active-passive GPFS cluster

In an active-passive environment, two GPFS clusters are set up in two geographically distinct locations (the production and the recovery sites). These clusters are referred to as peer GPFS clusters.

A GPFS file system is defined over a set of disk volumes located at the production site and these disks are mirrored using storage replication to a secondary set of volumes located at the recovery site. During normal operation, only the nodes in the production GPFS cluster mount and access the GPFS file system at any given time, which is the primary difference between a configuration of this type and the active-active model.

In the event of a catastrophe in the production cluster, the storage replication target devices are made available to be used by the nodes in the recovery site.

The secondary replica is then mounted on nodes in the recovery cluster as a regular GPFS file system, thus allowing the processing of data to resume at the recovery site. At a latter point, after restoring the physical operation of the production site, we execute the fallback procedure to resynchronize the content of the replicated volume pairs between the two clusters and re-enable access to the file system in the production environment.

The high-level organization of synchronous active-passive storage replication based GPFS cluster is shown in [Figure 21 on page 639](#).

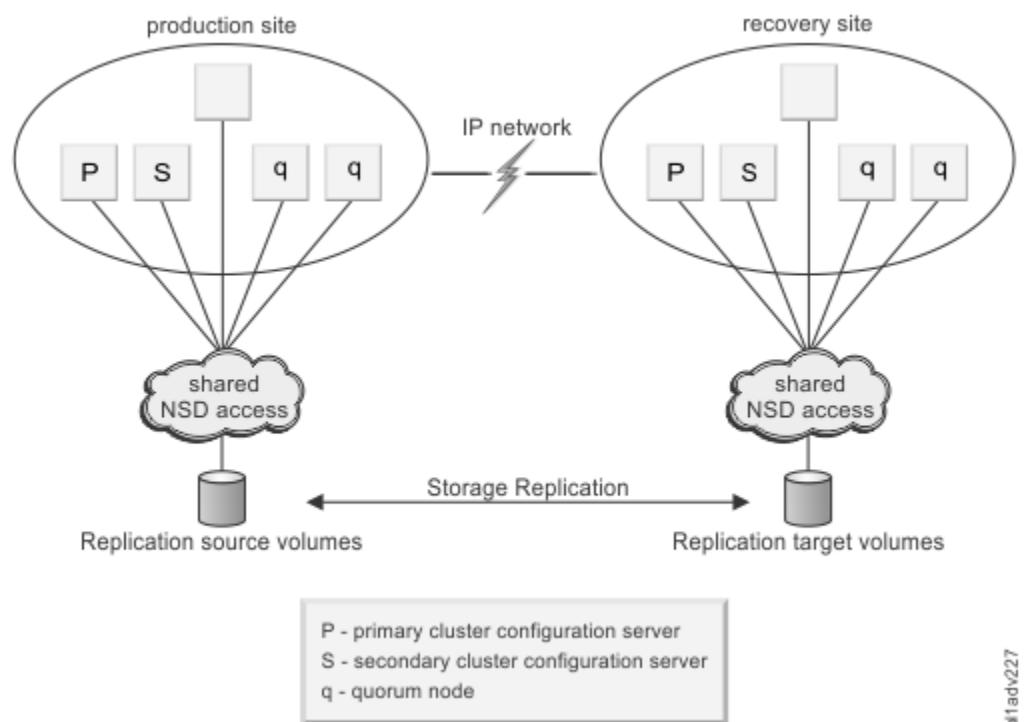


Figure 21. A synchronous active-passive storage replication-based GPFS configuration without a tiebreaker site

Setting up an active-passive GPFS configuration

This example demonstrates how to configure an active-passive GPFS cluster.

To establish an active-passive storage replication GPFS cluster as shown in [Figure 21 on page 639](#), consider the configuration:

Production site

Consists of:

- Nodes – **nodeP001, nodeP002, nodeP003, nodeP004, nodeP005**
- Storage subsystems – Storage System **P**

- LUN IDs and disk volume names – **lunP1 (hdisk11), lunP2 (hdisk12), lunP3 (hdisk13), lunP4 (hdisk14)**

Recovery site

Consists of:

- Nodes – **nodeR001, nodeR002, nodeR003, nodeR004, nodeR005**
- Storage subsystems – Storage System R
- LUN ids and disk volume names – **lunR1 (hdisk11), lunR2 (hdisk12), lunR3 (hdisk13), lunR4 (hdisk14)**

All disks are SAN-attached and directly accessible from all local nodes.

1. Establish synchronous PPRC volume pairs by using the **copy entire volume** option:

```

lunP1-lunR1 (source-target)
lunP2-lunR2 (source-target)
lunP3-lunR3 (source-target)
lunP4-lunR4 (source-target)

```

2. Create the recovery cluster selecting **nodeR001** as the primary cluster data server node, **nodeR002** as the secondary cluster data server nodes, and the nodes in the cluster contained in the file **NodeDescFileR**. The **NodeDescFileR** file contains the node descriptors:

```

nodeR001:quorum-manager
nodeR002:quorum-manager
nodeR003:quorum-manager
nodeR004:quorum-manager
nodeR005

```

Issue this command:

```
mmcrcluster -N NodeDescFileR -p nodeR001 -s nodeR002
```

3. Create the GPFS production cluster selecting **nodeP001** as the primary cluster data server node, **nodeP002** as the secondary cluster data server node, and the nodes in the cluster contained in the file **NodeDescFileP**. The **NodeDescFileP** file contains the node descriptors:

```

nodeP001:quorum-manager
nodeP002:quorum-manager
nodeP003:quorum-manager
nodeP004:quorum-manager
nodeP005

```

Issue this command:

```
mmcrcluster -N NodeDescFileP -p nodeP001 -s nodeP002
```

4. At all times the peer clusters must see a consistent image of the mirrored file system's configuration state contained in the **mmsdrfs** file. After the initial creation of the file system, all subsequent updates to the local configuration data must be propagated and imported into the peer cluster. Execute the **mmfsctl syncFSconfig** command to resynchronize the configuration state between the peer clusters after each of these actions in the primary GPFS cluster:

- Addition of disks through the **mmadddisk** command
- Removal of disks through the **mmdeldisk** command
- Replacement of disks through the **mmrpldisk** command
- Modifications to disk attributes through the **mmchdisk** command
- Changes to the file system's mount point through the **mmchfs -T** command

To automate the propagation of the configuration state to the recovery cluster, activate and use the **syncFSconfig** user exit. Follow the instructions in the prolog of **/usr/lpp/mmfs/samples/syncfsconfig.sample**.

5. From a node in the production cluster, start the GPFS daemon on all nodes:

```
mmstartup -a
```

6. Create the NSDs at the production site. The disk descriptors contained in the file **DiskDescFileP** are:

```
/dev/hdisk11:nodeP001:nodeP002:dataAndMetadata:-1  
/dev/hdisk12:nodeP001:nodeP002:dataAndMetadata:-1  
/dev/hdisk13:nodeP001:nodeP002:dataAndMetadata:-1  
/dev/hdisk14:nodeP001:nodeP002:dataAndMetadata:-1
```

Issue this command:

```
mmcinsd -F DiskDescFileP
```

7. Create the GPFS file system and mount it on all nodes at the production site:

```
mmcrfs /gpfs/fs0 fs0 -F DiskDescFileP
```

Failover to the recovery site and subsequent failback for an active-passive configuration

For an active-passive storage replication based cluster, complete these steps to fail over production to the recovery site.

Procedure when the Clustered Configuration Repository (CCR) scheme is in use

1. Stop the GPFS daemon on the surviving nodes as site **B** where the file `gpfs.siteB` lists all of the nodes at site **B**:

```
mmshutdown -N gpfs.siteB
```

2. Perform the appropriate commands to make the secondary replication devices available and change their status from being secondary devices to suspended primary devices.
3. If site **C**, the tiebreaker, failed along with site **A**, existing node quorum designations must be relaxed in order to allow the surviving site to fulfill quorum requirements. To relax node quorum, temporarily change the designation of each of the failed quorum nodes to nonquorum nodes using the `-- force` option:

```
mmchnode --nonquorum -N nodeA001, nodeA002, nodeA003, nodeC --force
```

4. Ensure that the source volumes are *not* accessible to the recovery site:

- Disconnect the cable
- Define the **nsdddevices** user exit file to exclude the source volumes

5. Restart the GPFS daemon on all surviving nodes:

```
mmstartup -N gpfs.siteB
```

Note: Make no further changes to the quorum designations at site B until the failed sites are back online and the following failback procedure has been completed. Do not shut down the current set of nodes on the surviving site B and restart operations on the failed sites A and C. This will result in a non-working cluster.

Failback procedure

After the physical operation of the production site has been restored, complete the failback procedure to transfer the file system activity back to the production GPFS cluster. The failback operation is a two-step process:

1. For each of the paired volumes, resynchronize the pairs in the reserve direction with the recovery LUN **lunRx** acting as the sources for the production LUN **lunPx**. An incremental resynchronization will

be performed which identifies the mismatching disk tracks, whose content is then copied from the recovery LUN to the production LUN. Once the data has been copied and the replication is running in the reverse direction this configuration can be maintained until a time is chosen to switch back to site **P**.

2. If the state of the system configuration has changed, update the GPFS configuration data in the production cluster to propagate the changes made while in failover mode. From a node at the recovery site, issue:

```
mmfsctl all syncFSconfig -n gpfs.sitePnodes
```

3. Stop GPFS on all nodes in the recovery cluster and reverse the disk roles so the original primary disks become the primaries again:

- a. From a node in the recovery cluster, stop the GPFS daemon on all nodes in the recovery cluster:

```
mmshutdown -a
```

- b. Perform the appropriate actions to switch the replication direction so that **diskA** is now the source and **diskB** is the target.

- c. From a node in the production cluster, start GPFS:

```
mmstartup -a
```

- d. From a node in the production cluster, mount the file system on all nodes in the production cluster.

Point-in-time copy of IBM Storage Scale data

Most storage systems provide the functions to make a point-in-time copy of data as an online backup mechanism. This function provides an instantaneous copy of the original data on the target disk, while the actual copy of data takes place asynchronously and is fully transparent to the user.

Several uses of the FlashCopy replica after its initial creation can be considered. For example, if your primary operating environment suffers a permanent loss or a corruption of data, you can choose to flash the target disks back onto the originals to quickly restore access to a copy of the file system as seen at the time of the previous snapshot. Before you restore the file system from a FlashCopy, make sure to suspend the activity of the GPFS client processes and unmount the file system on all GPFS nodes. FlashCopies also can be used to create a copy of data for disaster recovery testing and in this case are often taken from the secondary devices of a replication pair.

When a FlashCopy disk is first created, the subsystem establishes a control bitmap that is used after to track the changes between the source and the target disks. When processing read I/O requests sent to the target disk, this bitmap is consulted to determine whether the request can be satisfied by using the target's copy of the requested block. If the track that contains the requested data has not yet been copied, the source disk is instead accessed and its copy of the data is used to satisfy the request.

To prevent the appearance of out-of-order updates, it is important to consider data consistency when using FlashCopy. When taking the FlashCopy image, all disk volumes that make up the file system must be copied so that they reflect the same logical point in time. Two methods can be used to provide for data consistency in the FlashCopy image of your GPFS file system. Both techniques guarantee the consistency of the FlashCopy image by the means of temporary suspension of I/O, but either can be seen as the preferred method depending on your specific requirements and the nature of your GPFS client application.

FlashCopy provides for the availability of the file system's on-disk content in another GPFS cluster. But in order to make the file system that is known and accessible, you must issue the `mmfsctl syncFSConfig` command to:

- Import the state of the file system's configuration from the primary location.
- Propagate all relevant changes to the configuration in the primary cluster to its peer to prevent the risks of discrepancy between the peer's `mmsdrfs` file and the content of the file system descriptor found in the snapshot.

It is suggested you generate a new FlashCopy replica immediately after every administrative change to the state of the file system. This eliminates the risk of a discrepancy between the GPFS configuration data that is contained in the **mmsdrfs** file and the on-disk content of the replica.

Using consistency groups for Point in Time Copy

This topic provides a description about using consistency groups for point in time copy mechanism in IBM Storage Scale.

The use of FlashCopy consistency groups provides for the proper ordering of updates, but this method does not by itself suffice to guarantee the atomicity of updates as seen from the point of view of the user application. If the application process is actively writing data to GPFS, the on-disk content of the file system may, at any point in time, contain some number of incomplete data record updates and possibly some number of in-progress updates to the GPFS metadata. These appear as partial updates in the FlashCopy image of the file system, which must be dealt before enabling the image for normal file system use. The use of metadata logging techniques enables GPFS to detect and recover from these partial updates to the file system's metadata. However, ensuring the atomicity of updates to the actual data remains the responsibility of the user application. Consequently, the use of FlashCopy consistency groups is suitable only for applications that implement proper mechanisms for the recovery from incomplete updates to their data.

The FlashCopy consistency group mechanism is used to freeze the source disk volumes at the logical instant at which their logical image appears on the target disk volumes. The appropriate storage system documentation should be consulted to determine how to invoke the Point in Time Copy with the consistency group option.

Assuming a configuration with:

- Storage subsystems – 1
- LUN ids and disk volume names – **lunS1 (hdisk11)**, **lunS2 (hdisk12)**, **lunT1**, **lunT2**

lunS1 and **lunS2** are the FlashCopy source volumes. These disks are SAN-connected and appear on the GPFS nodes as **hdisk11** and **hdisk12**, respectively. A single GPFS file system **fs0** has been defined over these two disks.

lunT1 and **lunT2** are the FlashCopy target volumes. None of the GPFS nodes have direct connectivity to these disks.

To generate a FlashCopy image using a consistency group, do the following step:

Run the **establish FlashCopy pair** task with the **freeze FlashCopy consistency group** option. Create the volume pairs:

```
lunS1 - lunT1      (source-target)
lunS2 - lunT2      (source-target)
```

Using file-system-level suspension for Point in Time Copy

The use of file-system-level suspension through the **mmfscctl** command prevents incomplete updates in the FlashCopy image and is the suggested method for protecting the integrity of your FlashCopy images. Issuing the **mmfscctl** command leaves the on-disk copy of the file system in a fully consistent state, ready to be flashed and copied onto a set of backup disks. The command instructs GPFS to flush the data buffers on all nodes, write the cached metadata structures to disk, and suspend the execution of all subsequent I/O requests.

1. To initiate file-system-level suspension, issue the **mmfscctl suspend** command.
2. To resume normal file system I/O, issue the **mmfscctl resume** command.

Assuming a configuration with:

- Storage subsystems – ESS 1; logical subsystem LSS 1
- LUN ids and disk volume names – **lunS1 (hdisk11)**, **lunS2 (hdisk12)**, **lunT1**, **lunT2**

lunS1 and **lunS2** are the FlashCopy source volumes. These disks are SAN-connected and appear on the GPFS nodes as **hdisk11** and **hdisk12**, respectively. A single GPFS file system **fs0** has been defined over these two disks.

lunT1 and **lunT2** are the FlashCopy target volumes. None of the GPFS nodes have direct connectivity to these disks.

To generate a FlashCopy image using file-system-level suspension:

1. From any node in the GPFS cluster, suspend all file system activity and flush the GPFS buffers on all nodes:

```
mmfsctl fs0 suspend
```

2. Run the **establish FlashCopy pair** task to create the following volume pairs:

```
lunS1 - lunT1      (source-target)  
lunS2 - lunT2      (source-target)
```

3. From any node in the GPFS cluster, resume the file system activity:

```
mmfsctl fs0 resume
```

Chapter 44. Implementing a clustered NFS environment on Linux

In addition to the traditional exporting of GPFS file systems using the Network File System (NFS) protocol, GPFS allows you to configure a subset of the nodes in the cluster to provide a highly-available solution for exporting GPFS file systems using NFS.

Note: Available on all IBM Storage Scale editions.

The participating nodes are designated as Cluster NFS (CNFS) member nodes and the entire setup is frequently referred to as CNFS or a CNFS cluster.

In this solution, all CNFS nodes export the same file systems to the NFS clients. When one of the CNFS nodes fails, the NFS serving load moves from the failing node to another node in the CNFS cluster. Failover is done using recovery groups to help choose the preferred node for takeover. For the NFS client node to experience a seamless failover, hard mounts must be used. The use of soft mounts will likely result in stale NFS file handle conditions when a server experiences a problem, even though CNFS failover will still be done.

Currently, CNFS is supported only in the Linux environment. For an up-to-date list of supported operating systems, specific distributions, and other dependencies, refer to the [IBM Storage Scale FAQ in IBM Documentation](#).

NFS monitoring

Every node in the CNFS cluster runs a separate GPFS utility that monitors GPFS, NFS, and networking components on the node. Upon failure detection and based on your configuration, the monitoring utility might invoke a failover.

While an NFS server is in a grace period, the NFS monitor sets the NFS state of the server to "Degraded".

Related concepts

NFS failover

As part of GPFS recovery, the CNFS cluster failover mechanism is invoked. It transfers the NFS serving load that was served by the failing node to another node in the CNFS cluster. Failover is done using recovery groups to help choose the preferred node for takeover.

NFS locking and load balancing

Clustered Network File System (CNFS) supports a failover of all of the node's load together (all of its NFS IP addresses) as one unit to another node. However, if no locks are outstanding, individual IP addresses can be moved to other nodes for load balancing.

CNFS network setup

In addition to one set of IP addresses for the GPFS cluster, a separate set of one or more IP addresses is required for CNFS serving.

CNFS setup

You can set up a clustered NFS environment within a GPFS cluster.

CNFS administration

There are some common CNFS administration tasks in this topic along with a sample configuration.

NFS failover

As part of GPFS recovery, the CNFS cluster failover mechanism is invoked. It transfers the NFS serving load that was served by the failing node to another node in the CNFS cluster. Failover is done using recovery groups to help choose the preferred node for takeover.

The failover mechanism is based on IP address failover. The CNFS IP address is moved from the failing node to a healthy node in the CNFS cluster. In addition, it guarantees NFS lock (NLM) recovery.

Failover processing may involve rebooting of the problem node. To minimize the effects of the reboot, it is recommended that the CNFS nodes be dedicated to that purpose and are not used to run other critical processes. CNFS node rebooting should not be disabled or the failover reliability will be severely impacted.

Related concepts

NFS monitoring

Every node in the CNFS cluster runs a separate GPFS utility that monitors GPFS, NFS, and networking components on the node. Upon failure detection and based on your configuration, the monitoring utility might invoke a failover.

NFS locking and load balancing

Clustered Network File System (CNFS) supports a failover of all of the node's load together (all of its NFS IP addresses) as one unit to another node. However, if no locks are outstanding, individual IP addresses can be moved to other nodes for load balancing.

CNFS network setup

In addition to one set of IP addresses for the GPFS cluster, a separate set of one or more IP addresses is required for CNFS serving.

CNFS setup

You can set up a clustered NFS environment within a GPFS cluster.

CNFS administration

There are some common CNFS administration tasks in this topic along with a sample configuration.

NFS locking and load balancing

Clustered Network File System (CNFS) supports a failover of all of the node's load together (all of its NFS IP addresses) as one unit to another node. However, if no locks are outstanding, individual IP addresses can be moved to other nodes for load balancing.

CNFS depends on the domain name server (DNS) for any automated load balancing of NFS clients among the NFS cluster nodes. Using the round-robin algorithm is highly recommended.

Related concepts

NFS monitoring

Every node in the CNFS cluster runs a separate GPFS utility that monitors GPFS, NFS, and networking components on the node. Upon failure detection and based on your configuration, the monitoring utility might invoke a failover.

NFS failover

As part of GPFS recovery, the CNFS cluster failover mechanism is invoked. It transfers the NFS serving load that was served by the failing node to another node in the CNFS cluster. Failover is done using recovery groups to help choose the preferred node for takeover.

CNFS network setup

In addition to one set of IP addresses for the GPFS cluster, a separate set of one or more IP addresses is required for CNFS serving.

CNFS setup

You can set up a clustered NFS environment within a GPFS cluster.

CNFS administration

There are some common CNFS administration tasks in this topic along with a sample configuration.

CNFS network setup

In addition to one set of IP addresses for the GPFS cluster, a separate set of one or more IP addresses is required for CNFS serving.

The GPFS cluster can be defined over an IPv4 or IPv6 network. The IP addresses specified for CNFS can also be IPv4 or IPv6. The GPFS cluster and CNFS are not required to be on the same version of IP, but IPv6 must be enabled on GPFS to support IPv6 on CNFS.

Note: CNFS also requires the daemon network interface to accept incoming SSH connections.

Related concepts

NFS monitoring

Every node in the CNFS cluster runs a separate GPFS utility that monitors GPFS, NFS, and networking components on the node. Upon failure detection and based on your configuration, the monitoring utility might invoke a failover.

NFS failover

As part of GPFS recovery, the CNFS cluster failover mechanism is invoked. It transfers the NFS serving load that was served by the failing node to another node in the CNFS cluster. Failover is done using recovery groups to help choose the preferred node for takeover.

NFS locking and load balancing

Clustered Network File System (CNFS) supports a failover of all of the node's load together (all of its NFS IP addresses) as one unit to another node. However, if no locks are outstanding, individual IP addresses can be moved to other nodes for load balancing.

CNFS setup

You can set up a clustered NFS environment within a GPFS cluster.

CNFS administration

There are some common CNFS administration tasks in this topic along with a sample configuration.

CNFS setup

You can set up a clustered NFS environment within a GPFS cluster.

To do this, follow these steps:

1. Designate a separate directory for the CNFS shared files:

```
mmchconfig cnfsSharedRoot=directory
```

where:

cnfsSharedRoot=directory

Is the path name to a GPFS directory, preferably on a small separate file system that is not exported by NFS. The GPFS file system that contains the directory must be configured to be mounted automatically upon GPFS start on each of the CNFS nodes (-A yes option on the mmchfs command). **cnfsSharedRoot** is a mandatory parameter and must be defined first. This directory must be readable by the **rpcuser** ID.

2. Add all GPFS file systems that need to be exported to **/etc(exports)**. For NSF export considerations, see the topic *Exporting a GPFS file system using NFS* in the *IBM Storage Scale: Administration Guide*.
3. If the shared directory from step 1 is in an exported file system, restrict access to that directory.
4. Use the mmchnode command to add nodes to the CNFS cluster:

```
mmchnode --cnfs-interface=ip_address_list -N node
```

where:

ip_address_list

Is a comma-separated list of host names or IP addresses to be used for GPFS cluster NFS serving.

node

Identifies a GPFS node to be added to the CNFS cluster.

For more information, see the topic *mmchnode command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

5. Use the mmchconfig command to configure the optional CNFS parameters.

cnfsMountdPort=mountd_port

Specifies the port number to be used for the **rpc.mountd** daemon.

For CNFS to work correctly with the automounter (AMD), the **rpc.mountd** daemon on the different nodes must be bound to the same port.

cnfsNFSprocs=nfsd_procs

Specifies the number of **nfsd** kernel threads. The default is 32.

cnfsVersions=nfs_versions

Specifies a comma-separated list of protocol versions that CNFS should start and monitor. The default is 3, 4. If you are not using NFS v3 and NFS v4, specify this parameter with the appropriate values for your configuration.

Note: If you are not using NFS v3 and NFS v4, and you do not explicitly specify **cnfsVersions** with the protocol versions on your system, the following message will continually appear in the **mmfs.log**:

```
Found NFS version mismatch between CNFS and current running config, check the OS config files.
```

6. If multiple failover groups are desired, assign a group ID to each NFS node:

```
mmchnode --cnfs-groupid=groupid -N node
```

To assign NFS nodes to different groups, use a group ID that is in a different range of ten. For example, a node with group ID $2n$ will fail over only to nodes in the same range of ten (which means any node with group ID 20 to 29). Failover in the same group will first look for one of the nodes with the same group ID. If none are found, any node in the group range starting at $n0$ to $n9$ is selected.

Related concepts

NFS monitoring

Every node in the CNFS cluster runs a separate GPFS utility that monitors GPFS, NFS, and networking components on the node. Upon failure detection and based on your configuration, the monitoring utility might invoke a failover.

NFS failover

As part of GPFS recovery, the CNFS cluster failover mechanism is invoked. It transfers the NFS serving load that was served by the failing node to another node in the CNFS cluster. Failover is done using recovery groups to help choose the preferred node for takeover.

NFS locking and load balancing

Clustered Network File System (CNFS) supports a failover of all of the node's load together (all of its NFS IP addresses) as one unit to another node. However, if no locks are outstanding, individual IP addresses can be moved to other nodes for load balancing.

CNFS network setup

In addition to one set of IP addresses for the GPFS cluster, a separate set of one or more IP addresses is required for CNFS serving.

CNFS administration

There are some common CNFS administration tasks in this topic along with a sample configuration.

CNFS administration

There are some common CNFS administration tasks in this topic along with a sample configuration.

To query the current CNFS configuration, enter:

```
mm1scluster --cnfs
```

To temporarily disable CNFS on one or more nodes, enter:

```
mmchnode --cnfs-disable -N NodeList
```

Note: This operation affects only the high-availability aspects of the CNFS functionality. Normal NFS exporting of the data from the node is not affected. All currently defined CNFS IP addresses remain

unchanged. There will be no automatic failover from or to this node in case of a failure. If failover is desired, GPFS should be shut down on the affected node prior to issuing the mmchnode command.

To re-enable previously-disabled CNFS member nodes, enter:

```
mmchnode --cnfs-enable -N NodeList
```

Note: If the GPFS daemon is running on a node on which CNFS is being re-enabled, the node will try to activate its CNFS IP address. If the IP address was currently on some other CNFS-enabled node, that activation would include a takeover.

To permanently remove nodes from the CNFS cluster, enter:

```
mmchnode --cnfs-interface=DEFAULT -N NodeList
```

Note: This operation affects only the high-availability aspects of the CNFS functionality. Normal NFS exporting of the data from the node is not affected. All currently defined CNFS IP addresses remain unchanged. There will be no automatic failover from or to this node in case of a failure. If failover is desired, GPFS should be shut down on the affected node prior to issuing the mmchnode command.

A sample CNFS configuration

Here is a CNFS configuration example, which assumes the following:

- Your GPFS cluster contains three nodes: fin18, fin19, and fin20
- The host names for NFS serving are: fin18nfs, fin19nfs, and fin20nfs

To define a CNFS cluster made up of these nodes, follow these steps:

1. Add the desired GPFS file systems to **/etc(exports** on each of the nodes.
2. Create a directory called **ha** in one of the GPFS file systems by entering:

```
mkdir /gpfs/fs1/ha
```

3. Create a temporary file called **/tmp/hanfs-list**, which contains the following lines:

```
fin18 --cnfs-interface=fin18nfs  
fin19 --cnfs-interface=fin19nfs  
fin20 --cnfs-interface=fin20nfs
```

4. Set the CNFS shared directory by entering:

```
mmchconfig cnfsSharedRoot=/gpfs/fs1/ha
```

5. Create the CNFS cluster with the mmchnode command, by entering:

```
mmchnode -S /tmp/hanfs-list
```

6. Access the exported GPFS file systems over NFS. If one or more GPFS nodes fail, the NFS clients should continue uninterrupted.

Related concepts

NFS monitoring

Every node in the CNFS cluster runs a separate GPFS utility that monitors GPFS, NFS, and networking components on the node. Upon failure detection and based on your configuration, the monitoring utility might invoke a failover.

NFS failover

As part of GPFS recovery, the CNFS cluster failover mechanism is invoked. It transfers the NFS serving load that was served by the failing node to another node in the CNFS cluster. Failover is done using recovery groups to help choose the preferred node for takeover.

NFS locking and load balancing

Clustered Network File System (CNFS) supports a failover of all of the node's load together (all of its NFS IP addresses) as one unit to another node. However, if no locks are outstanding, individual IP addresses can be moved to other nodes for load balancing.

CNFS network setup

In addition to one set of IP addresses for the GPFS cluster, a separate set of one or more IP addresses is required for CNFS serving.

CNFS setup

You can set up a clustered NFS environment within a GPFS cluster.

Chapter 45. Implementing Cluster Export Services

Cluster Export Services (CES) provides highly available file and object services to a GPFS cluster by using Network File System (NFS), Object, or Server Message Block (SMB) protocols.

Note: Available on all IBM Storage Scale editions.

CES is an alternate approach to a clustered Network File System (CNFS) to export GPFS file systems. For more information about CES and protocol configuration, see [Chapter 3, “Configuring the CES and protocol configuration,” on page 49](#).

CES features

To successfully use Cluster Export Services (CES), you must consider function prerequisites, setup and configuration, failover/failback policies, and other management and administration requirements.

CES cluster setup

You can set up a Cluster Export Services (CES) environment within a GPFS cluster.

The CES shared root (cesSharedRoot) directory is needed for storing CES shared configuration data, protocol recovery, and some other protocol-specific purposes. It is part of the Cluster Export Configuration and is shared between the protocols. Every CES node requires access to the path that is configured as shared root.

To update the CES shared root directory, you must shut down the cluster, set the CES shared root directory, and start the cluster again:

```
mmshutdown -a  
mmchconfig cesSharedRoot=shared_root_path  
mmstartup -a
```

The recommendation for CES shared root directory is a dedicated file system. It can also reside in an existing GPFS file system. In any case, the CES shared root directory must be on GPFS and must be available when it is configured through the mmchconfig command.

To enable protocol nodes, the CES shared root directory must be defined. To enable protocol nodes, use the following command:

```
mmchnode --ces-enable -N Node1[,Node2...]
```

To disable a CES node, use the following command:

```
mmchnode --ces-disable -N Node1[,Node2...]
```

Preparing to perform service actions on the CES shared root directory file system

The CES shared root directory file system must be kept available for protocols operation to function. If a service action is to be performed on the CES shared root directory file system, perform the steps that follow.

Commands such as `mmshutdown`, `mmstartup` and `mmmount`, can be passed in the `cesnodes` node class parameter to ensure operation on all protocol nodes.

The following steps are used to perform service actions on the CES shared root file system:

1. Inform users of the impact to protocols. Quiesce protocol related I/O and mounts if possible. Quiesce cluster functions in progress on protocol nodes such as recalls, migrations, AFM, backup, and any policies that may be in use on the protocol nodes, or transition these cluster functions to other nodes. Finally, verify that file system quorum can be achieved by the remaining cluster nodes.

2. Shut down GPFS on all protocol nodes:

```
mmshutdown -N cesnodes
```

Note: Only protocol nodes need to be shut down for service of the CES shared root directory file system. However, other nodes may need to unmount the file system, depending on what service is being performed.

Protocol nodes are now ready for service actions to be performed on the CES shared root directory or the nodes themselves. To recover from a service action:

1. Start up GPFS on all protocol nodes:

```
mmstartup -N cesnodes
```

2. Make sure that the CES shared root directory file system is mounted on all protocol nodes:

```
mmmount cesSharedRoot -N cesnodes
```

3. Verify that all protocol services have been started:

```
mmces service list -a
```

Suspending or resuming CES nodes by using GUI

You can suspend or resume CES nodes from the **Services > CES Nodes** page of the IBM Storage Scale GUI.

Perform the following steps to suspend or resume CES nodes:

1. Go to **Services > CES Nodes** page in the IBM Storage Scale GUI.
2. To suspend a CES node, select the CES node and click **Suspend Node**. The Suspend Node window appears. Click **Suspend Node** in the Suspend node window to complete the process. When you suspend a CES node, you can also stop all the CES services on that node.
3. To resume a suspended CES node, select the node and click **Resume Node**. The Resume Node window appears. Click **Resume Node** in the Resume Node window to complete the process.

When the node is resumed, the CES IP addresses become active again. You can also start all the stopped service while performing the resume node operation.

You cannot enable or disable a CES service from the GUI. Use the **mmces** command in the CLI if you need to enable or disable a CES service in the cluster.

CES network configuration

Cluster Export Services (CES) IP addresses are used to export data via the NFS, SMB, and Object protocols. File and Object clients use the public IPs to access data on GPFS file systems.

CES IP addresses have the following characteristics:

- Shared between all CES protocols
- Organized in an *address pool* (there can be fewer or more CES addresses than nodes)
- Hosted on the CES nodes (there can be CES nodes without CES addresses)
- Can move between CES nodes (triggered manually via the command or as part of a CES failover)
- Must not be used for GPFS communication at the same time

CES IP addresses have these restrictions:

- The network on CES nodes must be configured so that all CES IPs can run on any CES node. Typically this configuration requires that all CES nodes have at least one NIC interface or VLAN-compatible interface with each CES IP network address. If different subnets are used, then all the CES IPs in a given CES group must be able to run on any node in that group.
- CES IPs are created as aliases on each CES node. Do not include the primary address of an adapter in the CES IP address pool.
- CES IPs must be resolvable by DNS or /etc/hosts.
- CES does not manage the subnet or netmask configuration. It is the user's task.

To add CES IP addresses to the address pool, use the mmces command:

```
mmces address add --ces-ip Address[,Address...]
```

By default, addresses are distributed among the CES nodes, but a new address can be assigned to a particular node:

```
mmces address add --ces-ip Address[,Address...] --ces-node Node
```

After a CES IP address is added to the address pool, you can manually move the address to a particular node:

```
mmces address move --ces-ip Address[,Address...] --ces-node Node
```

You can remove a CES IP address from the address pool with the mmces command:

```
mmces address remove --ces-ip Address[,Address...]
```

Removing an address while there are clients connected causes the clients to lose those connections. Any reconnection to the removed IP results in a failure. If DNS is used to map a name entry to one or more IP addresses, update the DNS to ensure that a client is not presented an address that was already removed from the pool. This process might also include invalidation of any DNS caches.

CES addresses are virtual IP addresses

The CES addresses that are assigned to the CES nodes are implemented as IP aliases. Each network adapter that hosts CES addresses must already be configured (with different non-CES IPs) in /etc/sysconfig. CES uses the netmask to figure out which interfaces to use. For example, if eth1 is 10.1.1.1 and eth2 is 9.1.1.1, then the CES IP 10.1.1.100 maps to eth1 and the CES IP 9.1.1.100 maps to eth2.

Virtual IP addresses include the following advantages:

- The node does not need to wait for the switch to accept the link when an IP address is failed back. Since the address is an alias, the interface on which it resides is already up.
- IP address failover is much faster.
- Administration is simplified by providing a clear distinction between the *system IP* and the *CES IP*. For example, you have a two-node cluster. One of the nodes in the two-node cluster has a problem that induces failover and someone logs in to the suspected node to reboot it. The surviving node might get rebooted by accident if the system address was migrated to the surviving node.

How to use an alias

To use an alias address for CES, you need to provide a static IP address that is not already defined as an alias in the /etc/sysconfig/network-scripts directory.

Before you enable the node as a CES node, configure the network adapters for each subnet that are represented in the CES address pool:

1. Define a static IP address for the device:

```
/etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE=eth1
```

```
BOOTPROTO=none
IPADDR=10.1.1.10
NETMASK=255.255.255.0
ONBOOT=yes
GATEWAY=10.1.1.1
TYPE=Ethernet
```

2. Ensure that there are no aliases that are defined in the network-scripts directory for this interface:

```
# ls -l /etc/sysconfig/network-scripts/ifcfg-eth1:*
ls: /etc/sysconfig/network-scripts/ifcfg-eth1:*: No such file or directory
```

After the node is enabled as a CES node, no further action is required. CES addresses are added as aliases to the already configured adapters.

CES address failover and distribution policies

When a Cluster Export Services (CES) node leaves the GPFS cluster, any CES IP addresses that are assigned to that node are moved to CES nodes still within the cluster. Additionally, certain error conditions and administrative operations can cause a node to release its addresses to be reassigned to other nodes.

As CES nodes enter and leave the GPFS cluster, the addresses are distributed among the nodes according to the address distribution policy that is selected. In addition, you can disable automatic address distribution to allow the user to manually maintain the address-to-node assignments.

The address distribution policy is set with the **mmces** command:

```
mmces address policy [even-coverage | balanced-load | node-affinity | none]
```

The following list describes each type of address distribution policy:

even-coverage

Distributes the addresses among the available nodes. The even-coverage policy is the default address distribution policy.

Note: If you have multiple CES networks, even IP address distribution in each network for every node might not be considered. The overall number of IP addresses on each node or CES group takes precedence.

Specify **mmces address move** to manually move IP addresses from one node to another node.

balanced-load

Distributes the addresses to approach an optimized load distribution. The loads (network and CPU) on all the nodes are monitored. Addresses are moved based on given policies for optimized load throughout the cluster.



Attention: In some cases, it is possible that balanced-load does not create an optimal load distribution. For robust balancing, use even-coverage load distribution that balances the number of CES-IPs as evenly as possible. IP address movements are minimized and IP address distribution might not be even.

node-affinity

Attempts to keep an address on the node to which the user manually assigned it. If the **mmces address add** command is used with the **--ces-node** option, the address is marked as being associated with that node. Similarly, if an address is moved with the **mmces address move** command, the address is marked as being associated with the destination node. Any automatic movement, such as reassigning a down node's addresses, does not change this association. Addresses that are enabled with no node specification do not have a node association.

Addresses that are associated with a node but assigned to a different node are moved back to the associated node if possible.

Important: Run the following command to view the node affinity preferred nodes:

```
mmces address list --full-list
```

If a CES IP address does not have the preferred affinity, you can move the CES IP address to the specified affinity node by running this command:

```
mmces address move
```

If you manually move the IP address by specifying `mmces address move`, it sets the affinity of all moved CES IP addresses to the target node.

Automatic address distribution is performed in the background in a way as to not disrupt the protocol servers more than necessary. If you want immediate redistribution of the addresses, use the `mmces` command to force an immediate rebalance:

```
mmces address move --rebalance
```

In order to prevent an interruption in service, IP addresses that have attributes assigned to them (for example: **object_database_node** or **object_singleton_node**) are not rebalanced.

IPs without a node assignment will not fail back if they were re-assigned due to some failure condition. You can run the following command to assign a node to an existing CES IP address:

```
mmces address move --ces-ip {IP[,IP...]} --ces-node NODE
```

You can run the following command to check the current assignment:

```
mmces address list --full-list
```

You can further control the assignment of CES addresses by placing nodes or addresses in CES groups. For more information, see the topic [“Configuring CES protocol service IP addresses” on page 52](#).

CES protocol management

Use the `mmces` command to enable or disable the Cluster Export Services (CES) protocols (NFS, SMB, Object, HDFS, S3, and Block).

Command examples:

```
mmces service enable {NFS | OBJ | SMB | HDFS | S3 | BLOCK}
```

```
mmces service disable {NFS | OBJ | SMB | HDFS | S3 | BLOCK} [ --force ]
```

After a protocol is enabled, the protocol is started on all CES nodes.

When a protocol is disabled, the protocol is stopped on all CES nodes and all protocol-specific configuration data is removed.

CES management and administration

Cluster Export Services (CES) nodes can be suspended for maintenance reasons with the `mmces` command.

For example:

```
mmces node suspend [-N Node[,Node...]]
```

When a node is suspended:

- The GPFS state of the node is unaffected. It remains an active member of the cluster.
- All CES IP addresses that are assigned to the node are reassigned to other nodes. Assignment of addresses to a suspend node is not allowed.
- All CES monitoring operations are stopped.
- Servers for the enabled CES protocols continue to run, but can be stopped.

- To unmount a GPFS file system used by NFS, the NFS server must be stopped and started on the node with the following mmces commands:

```
mmces service stop [NFS | OBJ | SMB | HDFS | BLOCK] [-N Node[,Node...]]  
mmces service start [NFS | OBJ | SMB | HDFS | BLOCK] [-N Node[,Node...]]
```

- A node or a group of nodes can be suspended and resume normal operation with the following mmces commands:

```
mmces node suspend -N node1,node2,node3  
mmces node resume
```

After a node is resumed, monitoring on the node is started and the node is eligible for address assignments.

CES NFS support

In Cluster Export Services (CES), you must consider supported protocol versions, service and export configuration, NFS service monitoring, fail-over considerations, and client support requirements for Network File System (NFS) support.

NFS support levels

NFS versions 3 (NFSv3) and 4 (NFSv4.0, NFSv4.1) are supported.

By default NFS 4.0 and NFS 4.1 versions are enabled on IBM Storage Scale 5.2.0 for the cluster installation. Clusters that are upgraded on any earlier version than IBM Storage Scale 5.2.0 retain the same default value (only NFS 4.0), or the value is set explicitly in older clusters (that is, only 4.0, only 4.1 or both 4.0 and 4.1). To modify the **MINOR_VERSION**, use the **mmnfs config change MINOR_VERSION=<minorversion>** command. Valid values for a minor version are 0 or 1 or 0,1. For more information, see the *mmnfs command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

NFS monitoring

The NFS servers are monitored to check for proper functions. If a problem is found, the CES addresses of the node are reassigned, and the node state is set to the failed state. When the problem is corrected, the node resumes normal operation.

NFS service configuration

Configuration options for the NFS service can be set with the **mmnfs config** command.

You can use the **mmnfs config** command to set and list default settings for NFS such as the port number for the NFS service, the default access mode for exported file systems, the log level, and enable or disable status for delegations. For a list of configurable attributes, see the *mmnfs command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Some of the attributes such as the protocol can be overruled for a given export on a per-client base. For example, the default settings might have NFS protocols 3 and 4 enabled, but the export for a client might restrict it to NFS version 4 only.

NFS export configuration

Exports can be added, removed, or changed with the **mmnfs export** command. Authentication must be set up before you define an export.

Exports can be declared for any directory in the GPFS file system, including a fileset junction. At the time where exports are declared, these folders must exist physically in GPFS. Only folders in the GPFS file system can be exported. No folders that are located only locally on a server node can be exported because they cannot be used in a failover situation.

Export-add and export-remove operations can be applied at run time of the NFS service. Export-change operation does not require a restart of the NFS service.

NFS failover

When a CES node leaves the cluster, the CES addresses assigned to that node are redistributed among the remaining nodes. Remote clients that access the GPFS file system might see a pause in service while the internal state information is passed to the new servers.

Note: NFS clients are responsible for maintaining data integrity when a server reboots, crashes, or fails over. In the NFS protocol, the NFS client is responsible for tracking which data is destaged and for detecting that a server is crashed before destaging all data, and for tracking which data must be rewritten to disk. Failover is transparent to most applications in NFS, with the following exception:

- Client applications might experience `-EEXIST` errors or `-ENOENT` errors when you are creating or deleting file system objects.

NFS clients

When you work with NFS clients, consider the following points:

- If you mount the same NFS export on one client from two different IBM Storage Scale NFS protocol nodes, data corruption might occur.
- The NFS protocol version that is used as the default on a client operating system might differ from what you expect. If you are using a client that mounts NFSv3 by default, and you want to mount NFSv4, then you must explicitly specify the relevant NFSv4.0 or NFSv4.1 in the `mount` command. For more information, see the `mount` command for your client operating system.
- To prevent NFS clients from encountering data integrity issues during failover, ensure that NFS clients are mounted with the option `-o hard`.
- A client can mount an NFS export by using one of the following methods:
 - CES IP of the protocol nodes.
 - Alias for the CES IPs, which are defined in the client's `/etc/hosts` path or DNS that includes DNS round-robin.
 - Net BIOS name as alias for CES IPs, which are defined in the client's `/etc/hosts` path or DNS that includes DNS round-robin. Net BIOS name can be found by using the `mmuserauth service list` command. This is mandatory if kerberized NFS shares are used.

Note:

- If protocol node hostname is used to mount NFS shares, then high availability for IP might not work.
- If a DNS round-robin (RR) entry name is used to mount an NFSv3 export, then data unavailability might occur due to unreleased locks. The NFS lock manager on IBM Storage Scale is not cluster-aware. This limitation is not applicable for NFSv4 exports.
- If the client mounts an NFS export by using a CES IP address, which is an IPv6 address, you might need to enclose the IPv6 address with square brackets. For example,

```
mount [spectrumScaleCESIPv6IP]:/path/to/exportedDirectory /localMountPoint
```

For more information about mounting with IPv6 address at the NFS client, see the man page for '`nfs`'.

- Clients that are performing NFS mounts must use a retry timeout value that is marginally lower than the NFS server grace period.

CES NFS server enters grace period after the daemon restarts, or when an IP address is released or a new IP address is acquired. Previously connected clients reclaim their state (for example – file locks, opens) within the grace period. The default value for grace period is 90 seconds.

The NFS client waits for a response from NFS server for a period that is indicated by `timeo` before retrying requests. The `timeo` can be specified as an option during `mount`. The value of `timeo` is

expressed in deciseconds (one-tenth of a second). Clients performing NFS mounts with a retry timeout value close to the NFS server grace period might cause application failures like I/O errors.

An example to set the retry timeout value as 40 seconds (overriding the Linux client's default value of 60 seconds for TCP) is - **mount -o timeo=400 spectrumScaleCESIP:/path/to/exportedDirectory /localMountPoint**.

Choosing between CNFS or CES

If you want to put highly available NFS services on top of the GPFS file system, you have the choice between clustered NFS ([Chapter 44, “Implementing a clustered NFS environment on Linux,” on page 645](#)) and Cluster Export Services ([Chapter 45, “Implementing Cluster Export Services,” on page 651](#)).

To help you choose one of these NFS offerings, consider the following points:

Multiprotocol support

If you plan to use other protocols (such as SMB or Object) in addition to NFS, CES must be chosen. While CNFS provides support only for NFS, the CES infrastructure adds support also for SMB and Object. With CES, you can start with NFS and add (or remove) other protocols at any time.

Command support

While CNFS provides native GPFS command support for creation and management of the CNFS cluster, it lacks commands to manage the NFS service and NFS exports. The CES infrastructure introduces native GPFS commands to manage the CES cluster. Furthermore, you can also manage the supported protocol services and the NFS exports by using the commands. For example, with CES, you do not need to adapt NFS configuration files individually on the protocol nodes. This work is done by the new GPFS commands that are provided for CES.

Performance

CNFS is based on the kernel NFS server while NFS support in CES is based on the Ganesha NFS server operating in user space. Due to the different nature of these NFS I/O stacks, performance depends on system characteristics and NFS workload. Contact your IBM representative to get help with sizing the required number of protocol nodes to support certain workload characteristics and protocol connection limits.

Which of the two NFS servers performs better has no general answer because the performance depends on many factors. Tests that are conducted with both NFS I/O stacks over various workloads show that the kernel-based NFS server (CNFS) performs better under metadata-intensive workloads. Typically this testing is with many smaller files and structures. The Ganesha NFS server provides better performance on other data-intensive workloads such as Video Streaming.

Note: CES provides a different interface to obtain performance metrics for NFS. CNFS uses the existing interfaces to obtain NFS metrics from the kernel (such as `nfsstat` or the `/proc` interface). The CES framework provides the `mmpfmon query` command for Ganesha-based NFS statistics. For more information, see the `mmpfmon` command topic in the *IBM Storage Scale: Command and Programming Reference Guide*.

Migration of CNFS to CES

For information about migrating existing CNFS environments to CES, see [“Migration of CNFS clusters to CES clusters” on page 662](#).

CES SMB support

In GPFS 4.1.1 and later, you can access a GPFS file system with an SMB client using its inherent SMB semantics.

The following features are provided:

Note: Some of the features described below require a higher version than 4.1.1.

Clustered SMB support

SMB clients can connect to any of the protocol nodes and get access to the shares defined. A clustered registry makes sure that all nodes see the same configuration data. Therefore, clients can connect to any Cluster Export Services (CES) node and see the same data. Moreover, the state of opened files (share modes, open modes, access masks, locks, and so on) is also shared among the CES nodes so that data integrity is maintained. On failures, clients can reconnect to another protocol node and IP addresses are transferred to another protocol node.

The supported protocol levels are SMB2 and the base functions of SMB3 (dialect negotiation, secure negotiation, encryption of data on the wire).

Export management command

With the **mmsmb** command, IBM Storage Scale provides a comprehensive entry point to manage all SMB-related configuration tasks like creating, changing, and deleting SMB shares.

SMB monitoring

The monitoring framework detects issue with the SMB services and triggers failover in case of an unrecoverable error.

Integrated installation

The SMB services are installed by the integrated installer together with the CES framework and the other protocols NFS and Object.

SMB performance metrics

The SMB services provide two sets of performance metrics that are collected by the performance monitor framework. Both current and historic data (with lower granularity) can be retrieved. The two sets of metrics are global SMB metrics (such as the number of connects and disconnects) and metrics for each SMB request (number, time, throughput). The **mpperfmon** query tool provides access to the most important SMB metrics via predefined queries. Moreover, metrics for the clustered file metadata database CTDB are collected and exposed via the **mpperfmon query** command.

Authentication and ID mapping

The SMB services can be configured to authenticate against the authentication services Microsoft Active Directory and LDAP. Mapping Microsoft security identifiers (SIDs) to the POSIX user and group IDs on the file server can either be done automatically by using the so-called autorid mechanism or external mapping services like RFC 2307 or Microsoft Services for Unix. If none of the offered authentication and mapping schemes matches the environmental requirements, a user-defined configuration can be established.

CES OBJ support

In Cluster Export Services (CES), you must consider several types of requirements for Object (OBJ) support.

OpenStack support levels

The Train release of OpenStack is used for Swift, Keystone, and their dependent packages. The Swift V1 and Keystone V3 APIs are also supported.

Object monitoring

The object servers are monitored to ensure that they function properly. If a problem is found, the CES addresses of the node are reassigned, and the node state is set to failed. When the problem is corrected, the node resumes normal operation.

Object service configuration

The Object service configuration is controlled by the respective Swift and Keystone configuration files. The master versions of these files are stored in the CCR repository, and copies exist in the /etc/swift and /etc/keystone directories on each protocol node. The files that are stored in those directories should not be directly modified since they are overwritten by the files that are stored in the CCR. To change the Swift or Keystone configuration, use the **mmobj config change** command to modify the master copy of configuration files stored in CCR. The monitoring framework is notified of the change and propagates the file to the local file system of the CES nodes. For information about the values that can be changed and their associated function, refer to the administration guides for Swift and Keystone.

To change the authentication that is used by the Keystone server, use the **mmuserauth** command to change the authentication repository to AD or LDAP, or to enable SSL communication to the Keystone server.

Object fileset configuration

A base fileset must be specified when the Object service is configured. An existing fileset can be used or a new fileset can be created. All filesets are created in the GPFS file system that is specified during installation. This fileset is automatically created in the GPFS file system that is specified during installation. Evaluate the data that is expected to be stored by the Object service to determine the required number of inodes that are needed. This expected number of inodes is specified during installation, but can be updated later by using standard GPFS file system and fileset management commands.

Object failover

When a CES node leaves the cluster, the CES addresses that are assigned to that node are redistributed among the remaining nodes. Remote clients that access the Object service might see active connections drop or a pause in service while the CES addresses are moved to the new servers. Clients with active connections to the CES addresses that are migrated might have their connections unexpectedly drop. Clients are expected to retry their requests when this happens.

Certain Object-related services can be migrated when a node is taken offline. If the node was hosting the backend database for Keystone or certain Swift services that are designated as singletons (such as the auditor), those services are started on the active node that received the associated CES addresses of the failed node. Normal operation of the Object service resumes after the CES addresses are reassigned and necessary services automatically restarted.

Object clients

The Object service is based on Swift and Keystone, and externalizes their associated interfaces. Clients should follow the associated specifications for those interfaces. Clients must be able to handle dropped connections or delays during CES node failover. In such situations, clients should retry the request or allow more time for the request to complete.

To connect to an Object service, clients should use a load balancer or DNS service to distribute requests among the pool of CES IP addresses. Clients in a production environment should not use hard-coded CES addresses to connect to Object services. For example, the authentication URL should refer to a DNS host name or a load balancer front end name such as `http://protocols.gpfs.net:35357/v3` rather than a CES address.

Inode Allocation Overview

Object storage consumes fileset inodes when the unified file and object access layout is used. One inode is used for each file or object, and one inode is used for each directory in the object path.

In the traditional object layout, objects are placed in the following directory path:

```
gpfs filesystem root/fileset/o/virtual device/objects/partition/hash_suffix/hash/object
```

An example object path is:

```
/ibm/gpfs/objfs/o/z1device111/objects/11247/73a/afbeca778982b05b9dddf4fed88f773a/1461036399.66296.data
```

Similarly, account and container databases are placed in the following directory paths:

```
gpfs filesystem root/fileset/ac/virtual device/containers/partition/hash_suffix/hash/account db and
```

```
gpfs filesystem root/fileset/ac/virtual device/containers/partition/hash_suffix/hash/container db.
```

An example account path is:

```
/ibm/gpfs/objfs/ac/z1device62/accounts/13700/f60/d61003e46b4945e0bbbfcee341d30f60/d61003e46b4945e0bbbfcee341d30f60.db
```

An example container path is:

```
/ibm/gpfs/objfs/ac/z1device23/containers/3386/0a9/34ea8d244872a1105b7df2a2e6ede0a9/34ea8d244872a1105b7df2a2e6ede0a9.db
```

Starting at the bottom of the object path and working upward, each new object that is created requires a new hash directory and a new object file, thereby consuming two inodes. Similarly, for account and container data, each new account and container require a new hash directory and a db file. Also, a db.pending and a lock file is required to serialize access. Therefore, four inodes are consumed for each account and each container at the hash directory level.

If the parent directories do not already exist, they are created, thereby consuming additional inodes. The hash suffix directory is three hexadecimal characters, so there can be a maximum of 0xFFFF or 4096 suffix directories per partition. The total number of partitions is specified during initial configuration. For IBM Storage Scale, 16384 partitions are allocated to objects and the same number is allocated to accounts and containers.

For each object partition directory, the hashes.pkl file is created to track the contents of the partition subdirectories. Also, there is a .lock file that is created for each partition directory to serialize updates to hashes.pkl. This is a total of three inodes required for each object partition.

There are 128 virtual devices allocated to object data during initial configuration, and the same number is allocated to account and container data. For each virtual device a tmp directory is created to store objects during upload. In the async_pending directory, container update requests that time out are stored until they are processed asynchronously by the object updater service.

The total number of inodes used for object storage in the traditional object layout can be estimated as follows:

```
total required inodes = account & container inodes + object inodes
```

As per this information, there are four inodes per account hash directory and four inodes per container hash directory. In the worst case, there would be one suffix directory, one partition directory, and one virtual device directory for each account and container. Therefore, the maximum inodes for accounts and containers can be estimated as:

```
account and container inodes = (7 * maximum number of accounts) + (7 * maximum number of containers)
```

In a typical object store there are more objects than containers, and more containers than accounts. Therefore, while estimating the required inodes, we estimate the number of inodes required for accounts and containers to be seven times the maximum number of containers. The maximum required inodes can be calculated as shown below:

```
max required inodes = (inodes for objects and hash directory) + (inodes required for hash
directories) +
(inodes required for partition directories and partition metadata) +
(inodes required for virtual devices) + (inodes required for containers)
max required inodes = (2 x maximum number of objects) + (4096 inodes per partition * 16384
partitions) +
(16384 partitions * 3) + (128 inodes) + (7 * maximum number of containers)
```

Important: As the number of objects grows, the inode requirement is dominated by the number of objects. A safe rule of thumb is to allocate three inodes per expected object when there are 10M to 100M expected objects. For more than 100M, you can allocate closer to 2.5 inodes per object.

Note: This applies to a case when all objects as well as account and container data are in the same fileset. While using multiple storage policy filesets or a different fileset for account and container data, the calculations must be adjusted.

CES HDFS support

Starting from IBM Storage Scale 5.0.4.2, CES also supports HDFS protocols. For more information, see [CES HDFS in Big data and analytics support documentation](#).

The following features are available:

HDFS overview

CES HDFS follows the same generic installation methods and prerequisites like the other protocols. For more information about CES HDFS, see the *Overview* and *Limitations and Recommendations* topics in [Big data and analytics support documentation](#).

Integrated installation

The HDFS services are installed by the integrated installer together with the CES framework and the other protocols NFS, SMB, and Object.

Management command

With the **mmhdfs** and **mmces** commands, IBM Storage Scale provides a comprehensive entry point to manage all HDFS-related configuration tasks.

HDFS monitoring

The monitoring framework detects HDFS Transparency name node and data node failures. The name node triggers a failover if an unrecoverable error occurs. For more information, see the *mmhealth*, *mmhdfs*, and the *mmces* commands in the *IBM Storage Scale: Command and Programming Reference Guide*.

Migration of CNFS clusters to CES clusters

If your system has established clustered Network File System (CNFS) clusters, you might consider the migration of these clusters to Cluster Export Services (CES) clusters.

Points to consider before you migrate

CES protocol nodes have the following dependencies and restrictions:

- CES nodes cannot coexist with CNFS clusters.

- The concepts of failover in CES node groups and CNFS failover groups are slightly different. While CNFS allows the failover not only within a group but also within ranges, CES does not. Make sure that your failover concepts are handled correctly by CES.
- CES nodes use SMB, NFS, and OpenStack SWIFT Object services.
- File system ACL permissions need to be in NFSv4 format.
- File system ACL semantics need to be set to NFSv4 format: `nfs4` ACL semantics in effect.
- CES SMB (Samba) services expect NFSv4 ACL formats.
- Existing CNFS exports definitions are not compatible with CES NFS. It is best to script and automate the creation of the equivalent exports by using the `mmnfs export add` command to reduce the amount you need to change in the future.
- CES nodes need authentication that is configured.
- A CES cluster has maximum 16 protocol nodes if the SMB protocol is also enabled.
- A CES cluster has maximum 32 protocol nodes if only NFS is enabled.

Because of a mutual exclusivity between CNFS and CES nodes, you need to accommodate user and application access outage while CES clusters nodes are installed, configured, set up for authentication, and the NFS exports are re-created. The duration of this process depends on the complexity of the customer environment.

You might want to procure new CES nodes or reuse the existing CNFS nodes. Either way, you cannot use the installation toolkit until the CNFS nodes are unconfigured.

If you could not test or plan the implementation of a CES cluster elsewhere, you might have to deal with the design and implementation considerations and issues during the planned outage period. Usually this process is straightforward and quick. If you have a more complex environment, it might take longer than the allotted upgrade window to complete the migration. In this case, it is possible to set up one or two non-CNFS, NFS servers to serve NFS for a short time. During this time, you would move all your CNFS IPs to these nodes as you decommission the CNFS cluster. Then, after you successfully set up your CES nodes, authentication, and corresponding exports, you can move the IPs from the temporary NFS servers over to the CES nodes.

Saving CNFS export configuration

You need to make a copy of the exports configuration file `/etc/exports` so that you can use this file as the basis for creating the new exports in CES NFS. CES NFS exports configuration needs to be created by using the `mmnfs export add` command or created in bulk by using the `mmnfs export load` command. When you unconfigure CNFS, you also need to delete the `/etc/exports` file from each of the CNFS nodes.

Steps to unconfigure CNFS

1. If you are planning to convert your existing CNFS nodes to CES nodes, see the support matrix first to know the supported configuration of a CES node. It is best to upgrade the nodes first while they are running CNFS. Because of these nodes upgrade, you can ensure that functions are the same as before you start to change over to CES nodes.
2. Ensure that you stop application and user access to the CNFS exports.
3. Run the `mmchnode` command to dereference or evict a CNFS node from the cluster. This command removes both the node and its associated IP:

```
# mmchnode -cnfs-interface=default -N node1Name,node2Name,...
```

4. When you remove the last node, CNFS is unconfigured and you see an output similar to this result:

```
[root@esnode3 ~]# mmlscluster --cnfs
GPFS cluster information
=====
GPFS cluster name: esvcluster1.esnode1
```

```
GPFS cluster id: 15635445795275488305
mmclscluster: CNFS is not defined in this cluster.
[root@esnode3 ~]#
```

5. Consider de-refencing the GPFS variable *cnfsSharedRoot*, although this step is not a requirement.
6. You can now delete the `/etc/exports` file on each of the CNFS nodes. Ensure that you have a backup copy of this file to use as a reference when you create the exports under CES NFS.
7. Run the **`systemctl disable nfs`** command to ensure kNFS does not start automatically.

Steps to Configure CES NFS

1. If you did not yet configure the CES nodes for authentication, complete this step before you create the exports. See “[CES NFS support](#)” on page 656 for details on configuring authentication for your environment.
2. Ensure that the file systems you want to export access to are configured for the NFSv4 security model. If you are converting an existing file system from another security model to NFSv4, you might need to review the ACL structures of the files and verify that your access works as expected.
3. If you have special configurations or options set in CNFS server, you might also want to reflect these settings in CES NFS. You need to review the appropriateness of these settings for the new environment. To change the settings, use the following command:

```
# mmnfs config change
```

4. If you have many exports to be converted to CES NFS, use the following command:

```
# mmnfs export load ExportCfgFile
```

ExportCfgFile contains a listing of all your exports as defined in the format that is used for `/etc/ganesha/gpfs.ganesha.exports.conf`.

5. Alternately, you can manually re-create each export on the CES cluster by using the **`mmnfs`** command.

```
# mmnfs export add Path --client ClientOptions
```

6. Before you proceed to configure CES nodes, remove the NFS exports from the `/etc/exports` file from each of the old CNFS nodes.
7. Add the IPs that were previously assigned to CNFS to the address pool to be managed by CES by using the following command:

```
# mmces address add --node nodeName --ces-ip ipAddress
```

See “[CES network configuration](#)” on page 652 for details about how to use this command.

8. Ensure that the IP addresses are unique and valid for your subnet.

For more information about creating protocol data exports, see *Fileset considerations for creating protocol data exports* in *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Test access to new exports on CES NFS

Test and verify that you have the same level of access to the NFS exports as you did on CNFS. This access test is needed to ensure that your applications and NFS clients can continue without further changes.

Chapter 46. Identity management on Windows / RFC 2307 attributes

GPFS allows file sharing among AIX, Linux, and Windows nodes. AIX and Linux rely on 32-bit user and group IDs for file ownership and access control purposes, while Windows uses variable-length security identifiers (SIDs). The difference in the user identity description models presents a challenge to any subsystem that allows for heterogeneous file sharing.

GPFS uses 32-bit ID namespace as the canonical namespace, and Windows SIDs are mapped into this namespace as needed. Two different mapping algorithms are used (depending on system configuration):

- GPFS built-in auto-generated mapping
- User-defined mappings are stored in the Microsoft Windows Active Directory by using the RFC 2307 attributes.

Auto-generated ID mappings

Auto-generated ID mappings are the default. If no explicit mappings are created by the system administrator in the Active Directory by using RFC 2307 attributes, all mappings between security identifiers (SIDs) and UNIX IDs will be created automatically by using a reserved range in UNIX ID space.

Note: If you have a mix of GPFS running on Windows and other Windows clients by accessing the integrated SMB server function, the ability to share data between these clients was not tested or validated. With protocol support, the SMB server might also be configured to automatically generate ID mapping. If you want to ensure that SMB users do not access data (share ID mapping) with Windows users, ensure that the automatic range for SMB server is different from this range. The range of IDs automatically generated for the SMB server can be controlled by `mmuserauth`.

Unless the default reserved ID range overlaps with an ID already in use, no further configuration is needed to use the auto-generated mapping function. If you have a specific file system or subtree that are only accessed by user applications from Windows nodes (even if AIX or Linux nodes are used as NSD servers), auto-generated mappings are sufficient for all application needs.

The default reserved ID range that is used by GPFS starts with ID 15,000,000 and covers 15,000,000 IDs. The reserved range should not overlap with any user or group ID in use on any AIX or Linux nodes. To change the starting location or the size of the reserved ID range, use the following GPFS configuration parameters:

sidAutoMapRangeLength

Controls the length of the reserved range for Windows SID to UNIX ID mapping.

sidAutoMapRangeStart

Specifies the start of the reserved range for Windows SID to UNIX ID mapping.

Note: For planning purposes, auto-generated ID mappings are stored permanently with file system metadata. A change in the **sidAutoMapRangeStart** value is only effective for file systems that are created after the configuration change.

Related concepts

[Configuring ID mappings in Active Directory Users and Computers for Windows Server 2016 \(and subsequent\) versions](#)

You can configure ID mappings in Active Directory Users and Computers (ADUC) for Windows Server 2016 (and subsequent) versions. You can also compare how IDMU attributes map to RFC 2307 attributes.

Related tasks

[Installing Windows IDMU](#)

The Identity Management for UNIX (IDMU) feature is included in Windows Server. This feature needs to be installed on the primary domain controller, as well as on any backup domain controllers. It is not

installed by default. There are two components that need to be installed in order for IDMU to function correctly. This applies to Windows Server 2012 R2 and preceding versions.

Configuring ID mappings in IDMU

Configuring ID mappings in Active Directory Users and Computers for Windows Server 2016 (and subsequent) versions

You can configure ID mappings in Active Directory Users and Computers (ADUC) for Windows Server 2016 (and subsequent) versions. You can also compare how IDMU attributes map to RFC 2307 attributes.

To configure ID mappings in Active Directory Users and Computers (ADUC) for Windows Server 2016 (and subsequent) versions, perform the following steps:

1. On the domain controller, click **Administrative Tools** and launch **Active Directory Users and Computers (ADUC)**.

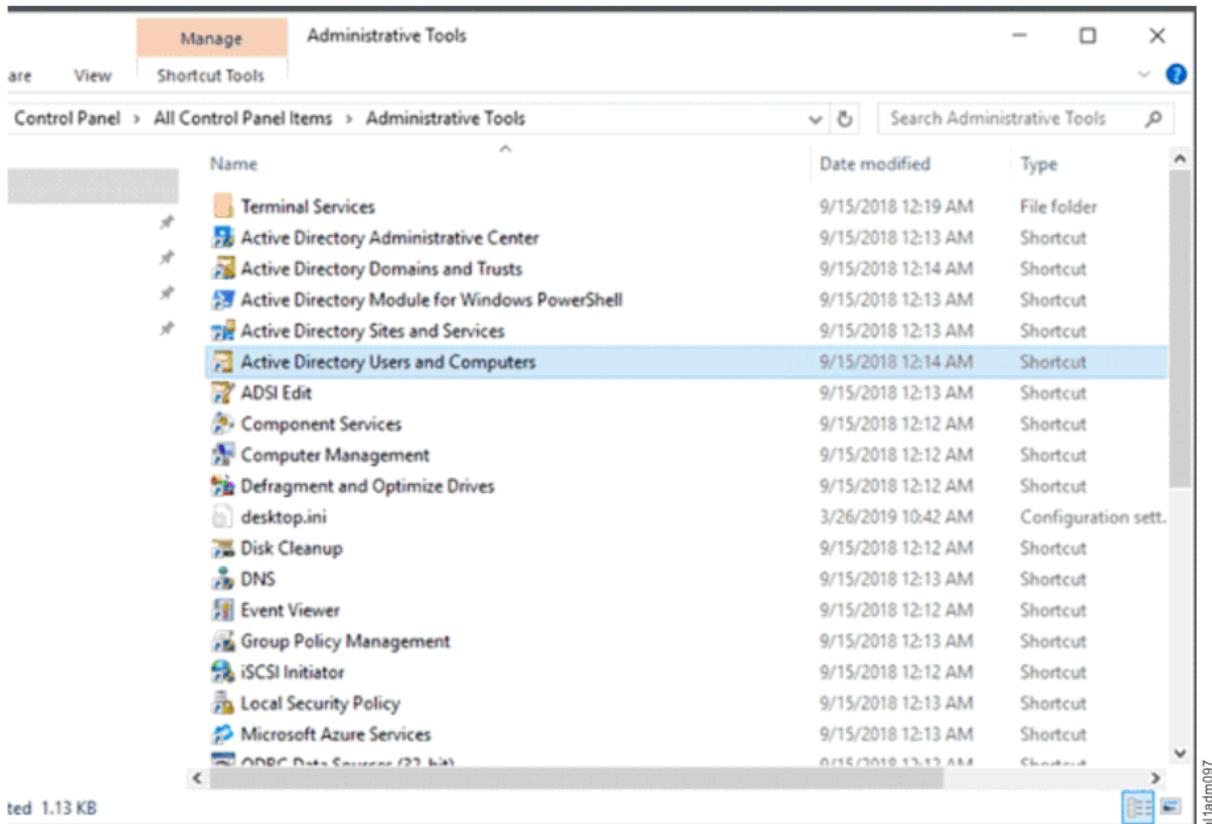


Figure 22. Opening the Active Directory Users and Computers directory

2. Enable **Advanced Features** from the **View** menu.

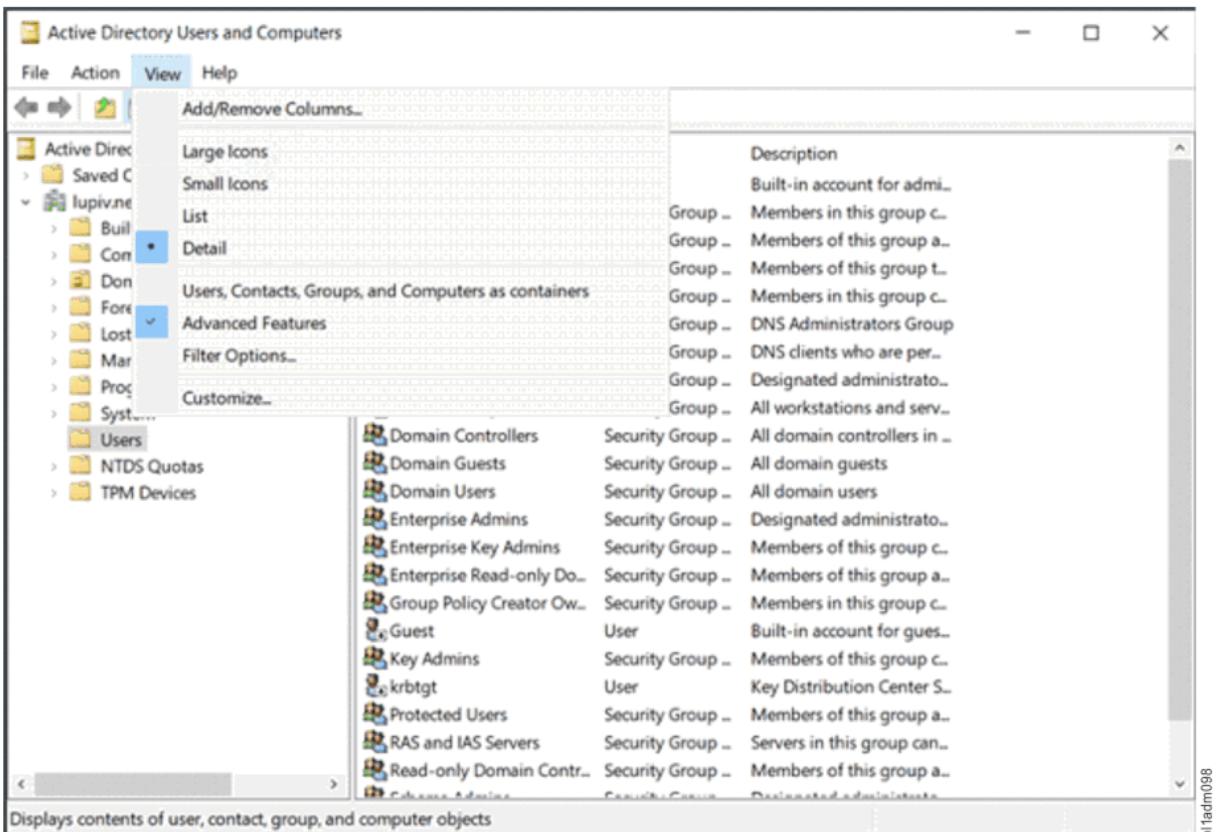


Figure 23. Enabling Advanced Features

3. Go to the specific user object under **Users**.

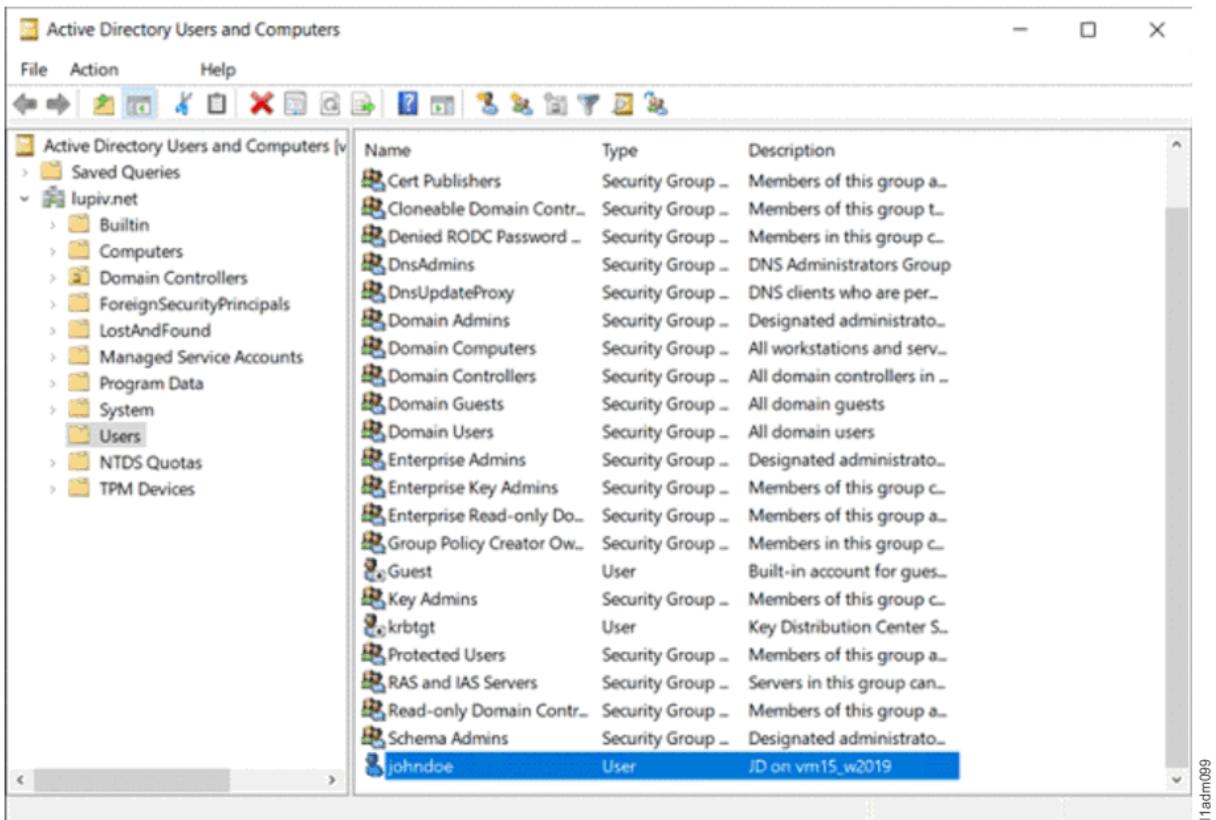


Figure 24. Accessing the user object in the *Users* directory

4. Right-click on **User object** to open the **Properties** menu, and then go to the **Attribute Editor** tab.

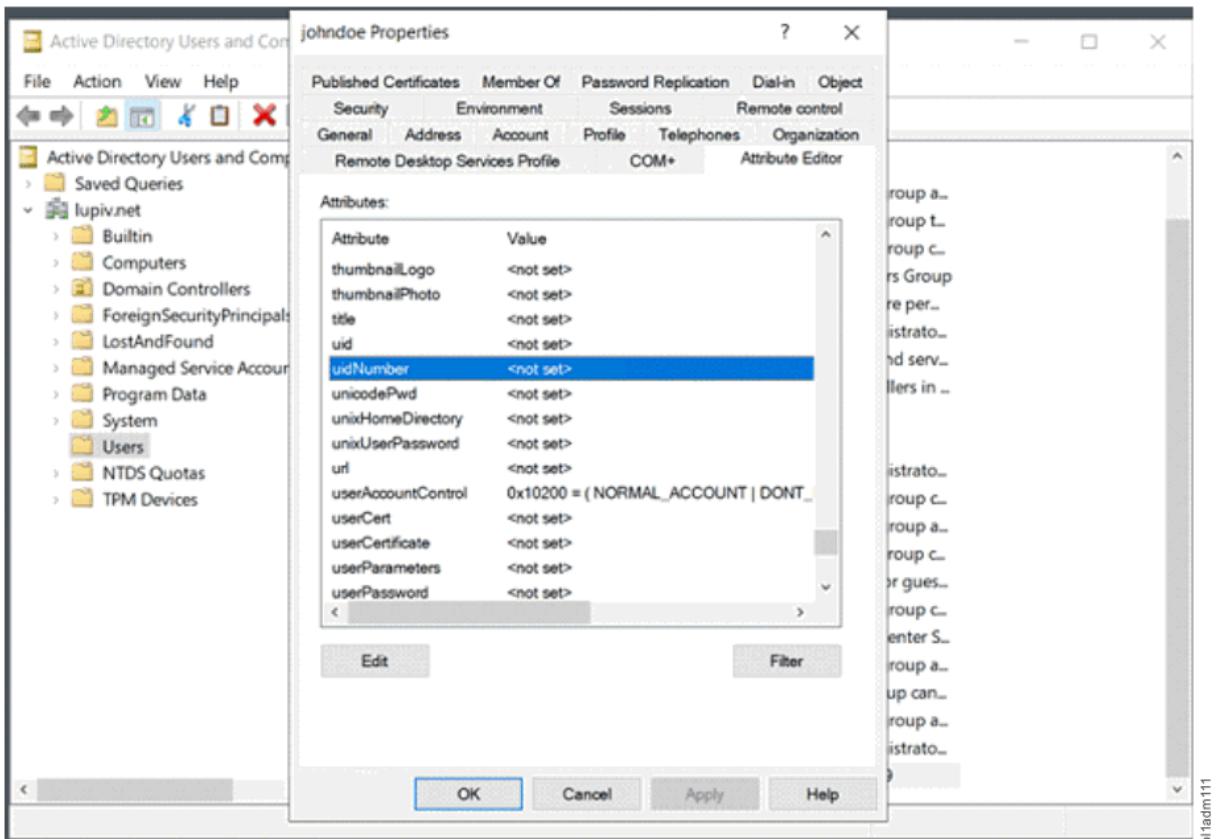


Figure 25. Displaying the user object properties

5. For users, specify the `uidNumber` attribute. For groups, specify the `gidNumber` attribute.

For information about how user information for Microsoft Identity Management for UNIX (IDMU) component UNIX attributes map to RFC 2307 attributes, use the following table:

Table 50. User identification attributes

Field on IMU Unix Attributes tab	RFC2307 AD attribute
UID	<code>uidNumber</code>
Logon Shell	<code>loginShell</code>
Home Directory	<code>unixHomeDirectory</code>
Primary group name or group identifier (GID)	<code>primaryGroupID</code>

For information about how groups information for Microsoft Identity Management for UNIX (IDMU) component UNIX attributes map to RFC 2307 attributes, use the following table:

Table 51. Group identification attribute

Field on IMU "Unix Attributes" tab	RFC2307 AD attribute
Group identifier (GID)	<code>gidNumber</code>

Related concepts

[Auto-generated ID mappings](#)

Auto-generated ID mappings are the default. If no explicit mappings are created by the system administrator in the Active Directory by using RFC 2307 attributes, all mappings between security identifiers (SIDs) and UNIX IDs will be created automatically by using a reserved range in UNIX ID space.

Related tasks

[Installing Windows IDMU](#)

The Identity Management for UNIX (IDMU) feature is included in Windows Server. This feature needs to be installed on the primary domain controller, as well as on any backup domain controllers. It is not installed by default. There are two components that need to be installed in order for IDMU to function correctly. This applies to Windows Server 2012 R2 and preceding versions.

[Configuring ID mappings in IDMU](#)

Installing Windows IDMU

The Identity Management for UNIX (IDMU) feature is included in Windows Server. This feature needs to be installed on the primary domain controller, as well as on any backup domain controllers. It is not installed by default. There are two components that need to be installed in order for IDMU to function correctly. This applies to Windows Server 2012 R2 and preceding versions.

Note: IDMU was deprecated in Windows Server 2012 and is not included in Windows Server 2016.

For more information, see *Configuring ID mappings in Active Directory Users and Computers for Active Directory Users and Computers (ADUC)* for instructions on editing RFC 2307 attributes in *IBM Storage Scale: Administration Guide*.

The only way to achieve Windows-Unix user-mapping in GPFS is with RFC 2307 attributes. These attributes can be administered by using Identity Mapping for Unix (IMU) from Microsoft in Windows Server versions up to and including Windows Server 2012 R2. Beginning Windows Server 2016, these RFC 2307 attributes can be specified by using the Active Directory Users and Computers (ADUC) MMC Snap-in.

To add the IDMU service when Active Directory is running on Windows Server 2008, follow these steps:

1. Open Server Manager.
2. Under **Roles**, select **Active Directory Domain Services**.
3. Under **Role Services**, select **Add Role Services**.
4. Under the **Identity Management for UNIX** role service, select **Server for Network Information Services**.
5. Click **Next**, then **Install**.
6. Restart the system when the installation completes.

Related concepts

[Auto-generated ID mappings](#)

Auto-generated ID mappings are the default. If no explicit mappings are created by the system administrator in the Active Directory by using RFC 2307 attributes, all mappings between security identifiers (SIDs) and UNIX IDs will be created automatically by using a reserved range in UNIX ID space.

[Configuring ID mappings in Active Directory Users and Computers for Windows Server 2016 \(and subsequent\) versions](#)

You can configure ID mappings in Active Directory Users and Computers (ADUC) for Windows Server 2016 (and subsequent) versions. You can also compare how IDMU attributes map to RFC 2307 attributes.

Related tasks

[Configuring ID mappings in IDMU](#)

Configuring ID mappings in IDMU

To configure ID mappings in Microsoft Identity Management for UNIX (IDMU), do the following steps. This procedure applies to Windows Server 2012 R2 and preceding versions.

These steps apply to Windows Server up to and including Windows Server 2012 R2 versions, which have IDMU. Because IDMU was removed starting Windows Server 2016, see *Configuring ID mappings in Active Directory Users and Computers for Active Directory Users and Computers (ADUC)* instructions on editing IDMU or RFC 2307 attributes in *IBM Storage Scale: Administration Guide*.

Typically it is a good idea to configure all the required ID mappings before you mount a GPFS file system for the first time. This configuration of ID mappings ensures that IBM Storage Scale stores only properly remapped IDs on the disk. However, you can add or delete ID mappings at any time while a GPFS file system is mounted. IBM Storage Scale checks the mapping changes every 60 seconds and uses updated mappings immediately.

When you configure an IDMU mapping for an ID that is already recorded in file metadata, you must be careful to avoid corrupting IDMU mappings and disrupting access to files. An auto-generated mapping that is already stored in an access control list (ACL) on disk continues to map correctly to a Windows SID. However, the SID is now mapped to a different UNIX ID. When you access a file with an ACL that contains the auto-generated ID, the access appears to IBM Storage Scale to be access by a different user. Depending on the file access permissions, the ID might not be able to access files that were previously accessible.

To restore proper file access for the affected ID, configure a new mapping and then rewrite the affected ACL. Rewriting replaces the auto-generated ID with an IDMU-mapped ID. To determine whether the ACL for a particular file contains auto-generated IDs or IDMU-mapped IDs, examine file ownership and permission information from a UNIX node, for example by issuing the `mmgetacl` command.

1. Click **Start > Administrative Tools > Active Directory Users and Computers**.
2. To see a list of the users and groups in this domain, select the **Users** branch in the tree on the left under the branch for your domain.
3. To open the **Properties** window for a user or group, double-click the user or group line.

If IDMU is set up correctly, the window includes a **UNIX Attributes** tab, as is shown in the following figure:

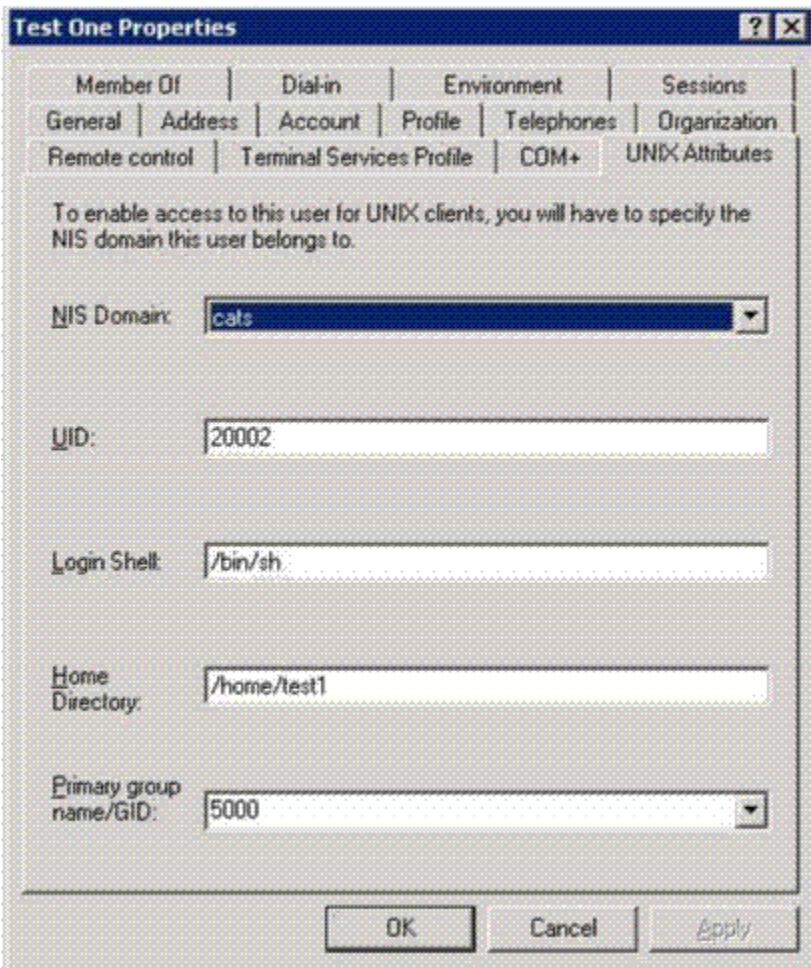


Figure 26. Properties window

4. To update information on the **UNIX Attributes** tab, do the following steps:
 - a) In the **NIS Domain** drop-down list, select the name of your Active Directory domain. To remove an existing mapping, click **<none>**.
Note: The field is labeled "NIS Domain" rather than just "Domain" because the IDMU subsystem was originally designed to support integration with the UNIX Network Information System (NIS). IBM Storage Scale does not use NIS.
 - b) In the **UID** field, enter a user ID. For group objects, enter a GID.
 Entering this information creates a bidirectional mapping between a UNIX ID and the corresponding Windows SID. To ensure that all mappings are unique, IDMU does not allow the same UID or GID for more than one user or group.
Note: You can create mappings for some built-in accounts in the **Builtin** branch of the **Active Directory Users and Computers** window.
 - c) Do not enter any information in the **Primary group name/GID** field. IBM Storage Scale does not use it.
5. To close the **Properties** window, click **OK**.

Related concepts

[Auto-generated ID mappings](#)

Auto-generated ID mappings are the default. If no explicit mappings are created by the system administrator in the Active Directory by using RFC 2307 attributes, all mappings between security identifiers (SIDs) and UNIX IDs will be created automatically by using a reserved range in UNIX ID space.

[Configuring ID mappings in Active Directory Users and Computers for Windows Server 2016 \(and subsequent\) versions](#)

You can configure ID mappings in Active Directory Users and Computers (ADUC) for Windows Server 2016 (and subsequent) versions. You can also compare how IDMU attributes map to RFC 2307 attributes.

Related tasks

[Installing Windows IDMU](#)

The Identity Management for UNIX (IDMU) feature is included in Windows Server. This feature needs to be installed on the primary domain controller, as well as on any backup domain controllers. It is not installed by default. There are two components that need to be installed in order for IDMU to function correctly. This applies to Windows Server 2012 R2 and preceding versions.

Chapter 47. Protocols cluster disaster recovery

Protocols cluster disaster recovery (DR) uses the capabilities of Active File Management (AFM) based Async Disaster Recovery (AFM DR) features to provide a solution that allows an IBM Storage Scale cluster to fail over to another cluster and fail back, and backup and restore the protocol configuration information in cases where a secondary cluster is not available.

Important: This feature is being deprecated.

For more information on AFM-based Async DR, see the topic *AFM-based Asynchronous Disaster Recovery (AFM DR)* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Important: Our initial feedback from the field suggests that success of a disaster recovery solution depends on administration discipline, including careful design, configuration and testing. Considering this, IBM has decided to disable the Active File Management-based Asynchronous Disaster Recovery feature (AFM DR) by default and require that customers deploying the AFM DR feature first review their deployments with IBM Storage Scale development. You should contact IBM Storage Scale Support at scale@us.ibm.com to have your use case reviewed. IBM will help optimize your tuning parameters and enable the feature. Please include this message while contacting IBM Support.

These limitations do not apply to base AFM support. These apply only to Async DR available with the IBM Storage Scale Advanced Edition 4.2 and 4.1.1.

For more information, see [Flash \(Alert\): IBM Storage Scale \(GPFS\) 4.2 and 4.1.1 AFM Async DR requirement for planning](#).

Protocols cluster disaster recovery limitations and prerequisites

For protocols cluster disaster recovery (DR) in an IBM Storage Scale cluster, the prerequisites and limitations are as follows.

Ensure that the following prerequisites are met for the secondary cluster for disaster recovery in an IBM Storage Scale with protocols.

- IBM Storage Scale is installed and configured.
- IBM Storage Scale code levels are the same on the primary and secondary clusters.
- IBM Storage Scale code levels are the same on every protocol node within a cluster.
- Cluster Export Services (CES) are installed and configured, and the shared root file system is defined.
- All protocols that are configured on the primary cluster are also configured on the secondary cluster.
- The authentication on the secondary cluster is identical to the authentication on the primary cluster.
- All exports that need to be protected by using AFM DR must have the same device and fileset name, and the same fileset link point on the secondary cluster as defined on the primary cluster.
- IBM NFSv3 stack must be configured on home cluster for the AFM DR transport of data.
- No data must be written to exports on secondary cluster while cluster is acting only as a secondary cluster before a failover.

The following limitations apply for disaster recovery in an IBM Storage Scale cluster with protocols.

- Only data that is contained within independent filesets can be configured for AFM-based Async Disaster Recovery (AFM DR). Therefore, all protocol exports that you want to be protected by DR must have the export path equal to the independent fileset link point.
- Backup and restore of the authentication configuration is not supported.
- After failover and failback or restore, all clients need to disconnect and then reconnect.
- If `--file-config --restore` is specified, perform the follow steps:
 - On failover: file authentication must be removed and then reconfigured on the secondary cluster.

- On restore: file authentication must be removed and then reconfigured on the primary cluster.
- On fallback: file authentication must be removed and then reconfigured on both primary and secondary clusters.
- IBM Storage Protect for Space Management and IBM Spectrum Archive migrated data within protocol exports is not supported within protocols cluster DR.
- IBM Storage Protect configuration file information is not automatically protected through protocols cluster DR.

Example setup for protocols disaster recovery

The following example scenario is used to show how to set up disaster recovery functionality for an IBM Storage Scale cluster with protocols.

This example consists of three NFS exports, three SMB shares, one object fileset, and two unified file and object access filessets that are also NFS exports. For the SMB and NFS exports, only two of each are independent filesets. This allows an AFM-based Async DR (AFM DR) configuration. For simplification, the filesets are named according to whether or not they were dependent or independent for the SMB and NFS exports. The inclusion of dependent filesets as exports is to show the warnings that are given when an export path is not an independent fileset link point.

NFS exports

- /gpfs/fs0/nfs-ganesha-dep
- /gpfs/fs0/nfs-ganesha1
- /gpfs/fs0/nfs-ganesha2

SMB shares

- /gpfs/fs0/smb1
- /gpfs/fs0/smb2
- /gpfs/fs0/smb-dep

Combination SMB and NFS exports

- /gpfs/fs0/combo1
- /gpfs/fs0/combo2

Setting up gateway nodes to ensure cluster communication during failover

Both the primary and the DR clusters require designating gateway nodes for access to whichever side is acting as the cache. By designating gateway nodes on both clusters, you can ensure that even during failover, cluster communication continues properly.

To handle a possible node failure, you need to specify at least two nodes on each cluster to be gateway nodes. To specify two nodes on the primary cluster as gateway nodes, use the command similar to the following:

mmchnode -N Node1,Node2 --gateway

Using the example setup mentioned in [“Example setup for protocols disaster recovery” on page 674](#), the command to specify gateway nodes on the primary cluster is as follows:

```
# mmchnode -N clusternode-vm1,clusternode-vm2 --gateway
Tue Apr 28 20:59:01 MST 2015: mmchnode: Processing node clusternode-vm2
Tue Apr 28 20:59:01 MST 2015: mmchnode: Processing node clusternode-vm1
mmchnode: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
# Tue Apr 28 20:59:04 MST 2015: mmccommon pushSdr_async:
mmsdrifs propagation started
```

```
Tue Apr 28 20:59:08 MST 2015: mmcommon pushSdr_async:  
mmsdrfs propagation completed; mmdsh rc=0
```

Similarly, you need to specify at least two nodes on the DR cluster as gateway nodes. Using the example setup, the command to specify gateway nodes on the DR cluster is as follows:

```
# mmchnode -N clusternode-vm1,clusternode-vm2 --gateway  
Tue Apr 28 20:59:49 MST 2015: mmchnode: Processing node clusternode-vm2  
Tue Apr 28 20:59:49 MST 2015: mmchnode: Processing node clusternode-vm1  
mmchnode: Propagating the cluster configuration data to all  
affected nodes. This is an asynchronous process.  
# Tue Apr 28 20:59:51 MST 2015: mmcommon pushSdr_async:  
mmsdrfs propagation started  
  
Tue Apr 28 20:59:54 MST 2015: mmcommon pushSdr_async:  
mmsdrfs propagation completed; mmdsh rc=0
```

Protocols and cluster configuration data required for disaster recovery

For protocols cluster disaster recovery, data needs to be collected for failover, fallback, backup, or restore from the respective protocol, for authentication, and for cluster wide information.

Use the following information to collect the data required for protocols cluster disaster recovery.

Swift Object data required for protocols cluster DR

Important:

- CES Swift Object protocol feature is not supported from IBM Storage Scale 5.2.0 onwards.
- IBM Storage Scale 5.1.8 is the last release that has CES Swift Object protocol.
- IBM Storage Scale 5.2.0 will tolerate the update of a CES node from IBM Storage Scale 5.1.8.
 - *Tolerate* means:
 - The CES node will be updated to 5.2.0.
 - Swift Object support will not be updated as part of the 5.2.0 update.
 - You may continue to use the version of Swift Object protocol that was provided in IBM Storage Scale 5.1.8 on the CES 5.2.0 node.
 - IBM will provide usage and known defect support for the version of Swift Object that was provided in IBM Storage Scale 5.1.8 until you migrate to a supported object solution that IBM Storage Scale provides.
 - CES Swift Object is replaced with IBM Storage Scale S3. For more details, refer the *S3 support overview* section the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
 - For more information about Swift Object in IBM Storage Scale, refer to the [IBM Storage Scale 5.2.0 documentation](#).
- Please contact IBM for further details and migration planning.

Related concepts

[SMB data required for protocols cluster DR](#)

Data required for the SMB protocol in case of a disaster recovery scenario is as follows.

[NFS data required for protocols cluster DR](#)

Data required for NFS protocol in case of a disaster recovery scenario is as follows.

[Authentication related data required for protocols cluster DR](#)

Authentication data that is necessary in a disaster recovery scenario is as follows.

[CES data required for protocols cluster DR](#)

Cluster Export Services (CES) data required in case of a disaster recovery scenario is as follows.

SMB data required for protocols cluster DR

Data required for the SMB protocol in case of a disaster recovery scenario is as follows.

You can determine the SMB shares using the **mmsmb export list** command.

The SMB protocol related files that need to be backed up are as follows.

- account_policy.tdb
- autorid.tdb₁
- group_mapping.tdb
- passdb.tdb
- registry.tdb
- secrets.tdb
- share_info.tdb
- ctdb.tdb

Note: ₁ This file is required only if the file authentication is configured with Active Directory.

The following information is common for all of these files:

- The type of these files is persistent TDB.
- They are not in the cluster configuration repository (CCR).
- Their location is /var/lib/ctdb/persistent on all protocol nodes.
- Their contents are same on all the nodes.
- Their name consists of TDB name + node specific extension. For example: registry.tdb.₀

The private Kerberos configuration files available at the following location also need to be backed up: /var/lib/samba/smb_krb5/. You can copy these files from this location and save them.

Related concepts

[Swift Object data required for protocols cluster DR](#)

[NFS data required for protocols cluster DR](#)

Data required for NFS protocol in case of a disaster recovery scenario is as follows.

[Authentication related data required for protocols cluster DR](#)

Authentication data that is necessary in a disaster recovery scenario is as follows.

[CES data required for protocols cluster DR](#)

Cluster Export Services (CES) data required in case of a disaster recovery scenario is as follows.

Failover steps for the SMB protocol

Use the following steps on a protocol node in the secondary cluster to fail over the SMB protocol configuration.

1. Before stopping the SMB services, make a note of the node number that will be used to restore TDB files because the node numbers are not available when SMB services are stopped.
2. Stop the SMB services using the following command:

```
mmces service stop SMB --all
```

3. Issue the following command to stop the NFS service:

```
mmces service stop NFS --all
```

4. Remove the contents of the /var/lib/ctdb/persistent directory on all protocol nodes.

5. Restore the previously saved TDB files to one of the protocol nodes and place them in the `/var/lib/ctdb/persistent` directory for the node where the node number was saved.

However, when copying the files to that directory on the node, replace the `X.bak`, where `X` represents the node number where the files were copied from, with the new node number. It is crucial that each of these files ends with `.tdb.Y`, where `Y` is the node number that was saved and the node number where the files are being restored. These files only need to be put into one of the nodes and when the SMB processes are started again they are copied around to the other nodes properly.

6. Remove the contents of the `/var/lib/samba/smb_krb5` directory on all the protocol nodes.
7. Restore the saved contents of `smb_krb5` to the `/var/lib/samba/smb_krb5/` directory on one of the protocol nodes. No special extension needs to be altered in this case.
8. Start the SMB services using the following command:

```
mmces service start SMB --all
```

9. Issue the following command to start the NFS service:

```
mmces service start NFS --all
```

10. Remove the SMB shares that are not protected using AFM DR independent filesets.

Fallback or restore steps for the SMB protocol

Use the following steps on a protocol node in the primary cluster, once repaired or replaced, to fail back or restore the SMB protocol configuration.

1. Stop the NFS services using the following command:

```
mmces service stop NFS -a
```

2. Stop the SMB services using the following command:

```
mmces service stop SMB -a
```

3. Delete all files from the `/var/lib/ctdb/persistent` directory on all protocol nodes.
4. Restore all required persistent TDB files from the saved configuration location to the `/var/lib/ctdb/persistent` directory on one of the protocol nodes. Ensure that you append the node number to the end of file names.
5. Delete all private Kerberos configuration files in the `/var/lib/samba/smb_krb5/` directory on all protocol nodes.
6. Restore private Kerberos configuration files to the `/var/lib/samba/smb_krb5/` directory on one of the protocol nodes.
7. On the fallback cluster, start the SMB services using the following command:

```
mmces service start SMB -a
```

8. Issue the following command on the fallback cluster to start the NFS service:

```
mmces service start NFS -a
```

NFS data required for protocols cluster DR

Data required for NFS protocol in case of a disaster recovery scenario is as follows.

To find the NFS exports, enter the following command:

```
mmnfs export list
```

The system displays output similar to the following:

Path	Delegations	Clients

```
/ibm/fs1/fset1 none      10.0.0.1  
/ibm/fs1/fset1 none      10.0.0.2  
/ibm/fs1/fset1 none      *
```

If the NFS exports are independent filesets, AFM based Disaster Recovery (AFM DR) can be used to replicate the data.

The NFS protocol related CCR files that need to be backed up are as follows.

- gpfs.ganesha.main.conf
- gpfs.ganesha.nfsd.conf
- gpfs.ganesha.log.conf
- gpfs.ganesha.exports.conf
- gpfs.ganesha.statdargs.conf

The following NFS protocol related CCR variable needs to be backed up.

- *nextexportid*

Related concepts

[Swift Object data required for protocols cluster DR](#)

[SMB data required for protocols cluster DR](#)

Data required for the SMB protocol in case of a disaster recovery scenario is as follows.

[Authentication related data required for protocols cluster DR](#)

Authentication data that is necessary in a disaster recovery scenario is as follows.

[CES data required for protocols cluster DR](#)

Cluster Export Services (CES) data required in case of a disaster recovery scenario is as follows.

Failover steps for the NFS protocol

Use the following steps on one of the protocol nodes in the secondary cluster to fail over the NFS protocol configuration.

1. Stop the NFS services by issuing the following command:

```
# mmces service stop NFS --all
```

2. Edit the saved `gpfs.ganesha.exports.conf` export configuration file to remove all exports that are not protected through AFM-DR independent filesets.
3. Restore the NFS-related CCR files. For a list of these files, see “[NFS data required for protocols cluster DR](#)” on page 677.
4. Restore the *nextexportid* CCR variable.
5. Load the exports file by issuing the following command:

```
# mnfs export load /Path_to_CCR_files/gpfs.ganesha.exports.conf
```

6. Start the NFS services by issuing the following command:

```
# mmces service start NFS --all
```

Fallback or restore steps for the NFS protocol

Use the following steps on a protocol node in the primary cluster, once repaired or replaced, to fail back or restore the NFS protocol configuration.

1. Stop the NFS services using the following command:

```
mmces service stop NFS --all
```

2. Restore the NFS related CCR files. For a list of these files, see “[NFS data required for protocols cluster DR](#)” on page 677.

3. Restore the *nextexportid* CCR variable.
4. Load the exports file using the following command:

```
mmnfs export load /<Path_to_saved_CCR_files>/gpfs.ganesha.exports.conf
```

5. Start the NFS services using the following command:

```
mmces service start NFS --all
```

Authentication related data required for protocols cluster DR

Authentication data that is necessary in a disaster recovery scenario is as follows.

The following authentication-related CCR file needs to be backed up for disaster recovery:

- authccr

File authentication-related data

The following CCR variable needs to be backed up for file authentication:

- FILE_AUTH_TYPE

Depending on the file authentication scheme that you are using, more files need to be backed up.

LDAP for file authentication:

- SSSD_CONF
- LDAP_CONF
- KRB5_CONF₁
- KRB5_KEYTAB₁
- LDAP_TLS_CACERT₁

Active Directory (AD) for file authentication:

- KRB5_CONF
- KRB5_KEYTAB₁

NIS for file authentication:

- SSSD_CONF
- YP_CONF

Note: ₁ This file is not always present.

Related concepts

[Swift Object data required for protocols cluster DR](#)

[SMB data required for protocols cluster DR](#)

Data required for the SMB protocol in case of a disaster recovery scenario is as follows.

[NFS data required for protocols cluster DR](#)

Data required for NFS protocol in case of a disaster recovery scenario is as follows.

[CES data required for protocols cluster DR](#)

Cluster Export Services (CES) data required in case of a disaster recovery scenario is as follows.

Failover steps for authentication data

Use the following steps on a protocol node in the secondary cluster to fail over the authentication configuration.

1. Save the current file authentication information on the secondary cluster.
2. Remove file authentication from the secondary cluster.

3. Restore file authentication on the secondary cluster based on the information that is saved in step 1.

Fallback steps for authentication data

Use the following steps on a protocol node in the primary cluster after the node is repaired or replaced to fail back the authentication configuration.

1. Save the current file authentication information on the primary cluster.
2. Remove file authentication from the primary cluster.
3. Restore file authentication on the primary cluster based on the information that is saved in step 1.

Restore steps for authentication data

Use the following steps on a protocol node in the primary cluster and the secondary cluster to restore the authentication configuration.

1. Save the current file authentication information on the primary cluster.
2. Remove file authentication from the primary cluster.
3. Restore file authentication on the primary cluster based on the information that is saved in step 1.
4. Save the current file authentication information on the secondary cluster.
5. Remove file authentication from the secondary cluster.
6. Restore file authentication on the secondary cluster based on the information that is saved in step 4.

CES data required for protocols cluster DR

Cluster Export Services (CES) data required in case of a disaster recovery scenario is as follows.

Cluster Configuration Repository (CCR) files that need to be backed up for CES in a disaster recovery scenario are as follows:

- `mmsdrfs`
- `cesiplist`

Related concepts

[Swift Object data required for protocols cluster DR](#)

[SMB data required for protocols cluster DR](#)

Data required for the SMB protocol in case of a disaster recovery scenario is as follows.

[NFS data required for protocols cluster DR](#)

Data required for NFS protocol in case of a disaster recovery scenario is as follows.

[Authentication related data required for protocols cluster DR](#)

Authentication data that is necessary in a disaster recovery scenario is as follows.

Failover steps for CES

- No Cluster Export Services (CES) configuration information is restored on fail over. This is because this information is typically cluster specific and it would interfere with the proper operating of the secondary cluster.

Fallback or recovery steps for CES

Use the following steps on a protocol node in the primary cluster to fail back or recover the CES configuration.

1. Restore the `cesiplist` file.
2. For each protocol node listed in the stored, backup copy of the `mmsdrfs` file, verify that the node on the primary cluster is also configured as a protocol node. If not, use the `mmchnode` to enable the node as a protocol node.

3. For each of the following CES parameters in the stored, backup copy of the `mmsdrfs` file, verify that the value is the same on the primary cluster. If not, use the `mmchconfig` to update the configuration value.

- `cesSharedRoot`
- `cesAddressPool`
- `cesServices`
- `cifsBypassTraversalChecking`
- `syncSambaMetadataOps`
- `cifsBypassShareLocksOnRename`

Chapter 48. File Placement Optimizer

A cluster in which all disks planned for IBM Storage Scale can be accessed only from one server (that means, no one disk could be accessed by 2 or more servers) is called as sharing nothing cluster.

For sharing nothing cluster, there are two typical configurations: replica-based IBM Storage Scale (sharing nothing cluster) and IBM Storage FPO.

If you do not run any workloads that could benefit from data locality (for example, SAP HANA + IBM Storage Scale for X86_64 machines, Hadoop, Spark, IBM DB2® DPF or IBM DashDB etc), you should not configure sharing nothing cluster as IBM Storage Scale FPO. For such workloads, you just need to configure replica-based IBM Storage Scale. Otherwise, you could configure it as IBM Storage Scale FPO (File Placement Optimizer). For IBM Storage Scale FPO, you could control the replica location in the file system.

When you create the storage pool over sharing nothing cluster, if you configure **allowWriteAffinity=yes** for the storage pool, you enable data locality for the data stored in the storage pool and this is called as FPO mode. If you configure **allowWriteAffinity=no** for the storage pool, this is called as replica-based sharing nothing mode. After the file system is created, the storage pool property **allowWriteAffinity** cannot be modified further.

In this chapter, all data locality related concepts (for example, allowWriteAffinity, Chunks, Extended failure groups, Write affinity failure group, Write affinity depth) are only effective for IBM Storage Scale FPO mode. For other concepts in this chapter, replica-based sharing nothing cluster is applicable.

Note: This feature is available with IBM Storage Scale Standard Edition or higher.

FPO uses the following entities and policies:

Chunks

A chunk is a logical grouping of blocks that allows the grouping to behave like one large block, useful for applications that need high sequential bandwidth. Chunks are specified by a block group factor that dictates how many file system blocks are laid out sequentially on disk to behave like a large block. Different Chunk size can be defined by block group factor on file level or defined globally on a storage pool by default.

On the file level, the block group factor can be specified by the `--block-group-factor` argument of the `mmchattr` command. You can also specify the block group factor by the `setBGF` argument of the `mmchpolicy` and `mmapplypolicy` command. The range of the block group factor is 1 - 1024. The default value is 1. You can also specify the block group factor through the `blockGroupFactor` argument in a storage pool stanza (as input to the `mmadddisk` or `mmcifs` command).

The effective chunk size is a multiplication of Block Group Factor and GPFS block size. For example, setting block size to 1 MB and block group factor to 128 leads to an effective large block size of 128 MB.

See the following command descriptions in the *IBM Storage Scale: Command and Programming Reference Guide*:

- `mmadddisk`
- `mmchattr`
- `mmcifs`
- `mmchpolicy`
- `mmapplypolicy`

Extended failure groups

A failure group is defined as a set of disks that share a common point of failure that might cause them all to become simultaneously unavailable. Traditionally, GPFS failure groups are identified by simple integers. In an FPO-enabled environment, a failure group might be specified as not just a single

number, but as a vector of up to three comma-separated numbers. This vector conveys topology information that GPFS exploits when making data placement decisions.

In general, a topology vector is a way for the user to specify which disks are closer together and which are farther away. In practice, the three elements of the failure group topology vector might represent the rack number of a disk, a position within the rack, and a node number. For example, the topology vector 2,1,0 identifies rack 2, bottom half, first node.

Also, the first two elements of the failure group represent the failure group ID and the three elements together represent the locality group ID. For example, 2,1 is the failure group ID and 2,1,0 is the locality group ID for the topology vector 2,1,0.

The Data block placement decisions about the disk selection for data replica are made by GPFS based on the Failure group. When considering two disks for striping or replica placement purposes, it is important to understand the following:

- Disks that differ in the first of the three numbers are farthest apart (as they are in different racks).
- Disks that have the same first number but differ in the second number are closer (as they are in the same rack, but in different halves).
- Disks that differ only in the third number reside in different nodes in the same half of the same rack.
- Only disks that have all three numbers in common reside in the same node.

The data block placement decisions are also affected by the level of replication and the value of the `writeAffinityDepth` parameter. For example, when using replication 3, GPFS might place two replicas far apart (different racks) to minimize chances of losing both. However, the third replica can be placed close to one of the others (same rack, but different half), to reduce network traffic between racks when writing the three replicas.

To specify the topology vector that identifies a failure group, you use the `failureGroup=FailureGroup` attribute in an NSD stanza (as input to the `mmadddisk` or `mmcrlfs` command).

See the following command descriptions in the *IBM Storage Scale: Command and Programming Reference Guide*:

- `mmadddisk`
- `mmcrlfs`

Write affinity depth

Write affinity depth is a policy that allows the application to determine the layout of a file in the cluster to optimize for typical access patterns. The write affinity is specified by a depth that indicates the number of localized copies (as opposed to wide striped). It can be specified at the storage pool or file level. The enabling of Write affinity depth, indicates that the first replica is being written on the node where the writing is triggered. It also indicates, the second and third replica (if any) are being written on the other node disks.

To specify write affinity depth, you use the `writeAffinityDepth` attribute in a storage pool stanza (as input to the `mmadddisk` or `mmcrlfs` command) or the `--write-affinity-depth` argument of the `mmchattr` command. You can use `--block-group-factor` argument of the `mmchpool` command to change a storage pool's block group factor. You can change write affinity depth by `--write-affinity-depth` argument of `mmchpool` for a storage pool. You can also specify the write affinity depth for file by the `setWAD` argument of the `mmchpolicy` and `mmapplypolicy` commands.

A write affinity depth of 0 indicates that each replica is to be striped across the disks in a cyclical fashion with the restriction that no two disks are in the same failure group. By default, the unit of striping is a block; however, if the block group factor is specified in order to exploit chunks, the unit of striping is a chunk.

A write affinity depth of 1 indicates that the first copy is written to the writer node. The second copy is written to a different rack. The third copy is written to the same rack as the second copy, but on a different half (which can be composed of several nodes).

A write affinity depth of 2 indicates that the first copy is written to the writer node. The second copy is written to the same rack as the first copy, but on a different half (which can be composed of several nodes). The target node is determined by a hash value on the fileset ID of the file, or it is chosen randomly if the file does not belong to any fileset. The third copy is striped across the disks in a cyclical fashion with the restriction that no two disks are in the same failure group. The following conditions must be met while using a write affinity depth of 2 to get evenly allocated space in all disks:

1. The configuration in disk number, disk size, and node number for each rack must be similar.
2. The number of nodes must be the same in the bottom half and the top half of each rack.

This behavior can be altered on an individual file basis by using the `--write-affinity-failure-group` option of the `mmchattr` command.

Note: In fileset level, Write affinity depth of 2 is design to assign (write) all the files in a fileset to the same second-replica node. However, this behavior depends on node status in the cluster. After a node is added to or deleted from a cluster, a different node might be selected as the second replica for files in a fileset.

See the description of storage pool stanzas that follows. Also, see the following command descriptions in the *IBM Storage Scale: Command and Programming Reference Guide*:

- `mmadddisk`
- `mmchattr`
- `mmcrls`
- `mmchpolicy`
- `mmapplypolicy`
- `mmchpool`

Write affinity failure group

Write affinity failure group is a policy that indicates the range of nodes (in a shared nothing architecture) where replicas of blocks in a particular file are to be written. The policy allows the application to determine the layout of a file in the cluster to optimize for typical access patterns.

You specify the write affinity failure group through the `write-affinity-failure-group WafgValueString` attribute of the `mmchattr` command. You can also specify write affinity failure group through the `setWADFG` attribute of the `mmchpolicy` and `mmapplypolicy` command. Failure group topology vector ranges specify the nodes, and the specification is repeated for each replica of the blocks in a file.

For example, the attribute `1,1,1:2;2,1,1:2;2,0,3:4` indicates:

- The first replica is on rack 1, rack location 1, nodes 1 or 2.
- The second replica is on rack 2, rack location 1, nodes 1 or 2.
- The third replica is on rack 2, rack location 0, nodes 3 or 4.

The default policy is a null specification. This default policy indicates that each replica must follow the storage pool or the file-write affinity depth (WAD) definition for data placement. Not wide striped over all disks.

When data in an FPO pool is backed up in the IBM Storage Protect server and then restored, the original placement map is broken unless you set the write affinity failure group for each file before backup.

Note: To change the failure group of a disk in a write-affinity-enabled storage pool, you must use the `mmdeldisk` and `mmadddisk` commands. You cannot use `mmchdisk` to change it directly.

See the following command descriptions in the *IBM Storage Scale: Command and Programming Reference Guide*:

- `mmchpolicy`
- `mmapplypolicy`

- mmchattr

Enabling the FPO features

To efficiently support write affinity and the rest of the FPO features, GPFS internally requires the creation of special allocation map formats. When you create a storage pool that is to contain files that make use of FPO features, you must specify `allowWriteAffinity=yes` in the storage pool stanza.

To enable the policy to read from preferred replicas, issue one of the following commands:

- To specify that the policy read from the first replica, regardless of whether there is a replica on the disk, default to or issue the following:

```
mmchconfig readReplicaPolicy=default
```

- To specify that the policy read replicas from the local disk, if the local disk has data, issue the following:

```
mmchconfig readReplicaPolicy=local
```

- To specify that the policy read replicas from the fastest disk to read from based on the disk's read I/O statistics, run the following:

```
mmchconfig readReplicaPolicy=fastest
```

Note: In an FPO-enabled file system, if you run data locality awareness workload over FPO, such as Hadoop or Spark, configure `readReplicaPolicy` as *local* to read data from the local disks to reduce the network bandwidth consumption.

See the description of storage pool stanzas that follows. Also, see the following command descriptions in the *IBM Storage Scale: Command and Programming Reference Guide*:

- mmadddisk
- mmchconfig
- mmcrlfs

Storage pool stanzas

Storage pool stanzas are used to specify the type of layout map and write affinity depth, and to enable write affinity, for each storage pool.

Storage pool stanzas have the following format:

```
%pool:
  pool=StoragePoolName
  blockSize=BlockSize
  usage={dataOnly | metadataOnly | dataAndMetadata}
  layoutMap={scatter | cluster}
  allowWriteAffinity={yes | no}
  writeAffinityDepth={0 | 1 | 2}
  blockGroupFactor=BlockGroupFactor
```

See the following command descriptions in the *IBM Storage Scale: Command and Programming Reference Guide*:

- mmadddisk
- mmcrlfs
- mmchpool

Recovery from disk failure

A typical shared nothing cluster is built with nodes that have direct-attached disks. Disks are not shared between nodes as in a regular GPFS cluster, so if the node is inaccessible, its disks are also inaccessible. GPFS provides means for automatic recovery from these and similar common disk failure situations.

The following command sets up and activates the disk recovery features:

```
mmchconfig restripeOnDiskFailure=yes -i
```

Usually, auto recovery must be enabled in an FPO cluster to protect data from multiple node failures. Set **mmchconfig restripeOnDiskFailure=yes -N all**. However, if one file system has only two failure groups for metadata or data with default replica two, or if one file system has only 3 failure groups for metadata or data with default replica 3, auto recovery must be disabled (**mmchconfig restripeOnDiskFailure=no -N all**) for IBM Storage Scale 4.1.x, 4.2.x and 5.0.0. The issue is fixed from IBM Storage Scale 5.0.1.

Whether a file system went through a recovery is determined by the max replication values for the file system. If the **mm1sfs -M** or **-R** value is greater than one, then the recovery code is run. The recovery actions are asynchronous and GPFS continues its processing while the recovery attempts take place. The results from the recovery actions and any errors that are encountered are recorded in the GPFS logs.

Two more parameters are available for fine-tuning the recovery process:

```
mmchconfig metadataDiskWaitTimeForRecovery=seconds  
mmchconfig dataDiskWaitTimeForRecovery=seconds
```

The default value for **metadataDiskWaitTimeForRecovery** is 1800 seconds. The default value for **dataDiskWaitTimeForRecovery** is 3600 seconds.

See the following command description in the *IBM Storage Scale: Command and Programming Reference Guide*:

- **mmchconfig**

Distributing data across a cluster

You can distribute data uniformly across a cluster.

You can distribute the data in the following possible ways:

- To ensure that the data is distributed evenly across all failure groups and all nodes within a failure group, import the data through a node that does not have any attached NSD and takes the role as a GPFS client node in the cluster.
- Use a write affinity depth of 0 across the cluster.
- Make every GPFS node an ingest node and deliver data equally across all the ingest nodes. However, this strategy is expensive in terms of implementation.

Ideally, all the failure groups must have an equal number of disks with roughly equal capacity. If one failure group is much smaller than the rest, it is likely to fill up faster than the others, therefore complicating the rebalancing actions.

After the initial ingesting of data, the cluster might be unbalanced. In such a situation, use the **mmrestripefs** command with the **-b** option to rebalance the data.

Note: For FPO users, the **mmrestripefs -b** command breaks the original data placement that follows the data locality rule.

FPO pool file placement and AFM

For AFM home or cache, an FPO pool file that is written on the local side is placed according to the write affinity depth and write affinity failure group definitions of the local side.

When a file is synced from home to cache, it follows the same FPO placement rule as when written from the gateway node in the cache cluster. When a file is synced from cache to home, it follows the same FPO data placement rule as when written from the NFS server in the home cluster.

To retain the same file placement at AFM home and cache, ensure that each has the same cluster configuration and set the write affinity failure group for each file. If the home and cache cluster have

different configurations, such as the disk number, node number, or fail group, then the data locality might be broken.

Configuring FPO

Follow the steps listed in the *IBM Storage Scale: Concepts, Planning, and Installation Guide* to install the IBM Storage Scale RPMs and build the portability layer on all nodes in the cluster.

You can configure password-less SSH for root user across all the IBM Storage Scale nodes. However, in cases of special security control, you can configure at least one node for the root user to access all the IBM Storage Scale nodes in a password-less mode. IBM Storage Scale commands can be run only over these nodes.

For OS with Linux kernel 2.6, enter the following commands on all the IBM Storage Scale nodes that are set as root to set **vm.min_free_bytes**:

```
# TOTAL_MEM=$(cat /proc/meminfo | grep MemTotal | tr -d \"[:alpha:]\" | tr -d \"[:punct:]\" | tr -d \"[:blank:]\") # VM_MIN_FREE_KB=$((TOTAL_MEM*6/100))

# echo "vm.min_free_kbytes = $VM_MIN_FREE_KB" >> /etc/sysctl.conf # sysctl -p

# sysctl -a | grep vm.min_free_kbytes
```

Configuring IBM Storage Scale Clusters

You must run all the IBM Storage Scale configuration steps as a root user. These steps must be executed only on one node.

Create the IBM Storage Scale Cluster

The IBM Storage Scale node file defines all nodes in the cluster and some of the roles.

Create the IBM Storage Scale cluster with **node11** as the primary and **node21** as the secondary cluster configuration server. Set the -A flag to automatically start GPFS daemons when the OS is started.

```
# mmcrcluster -A -C gpfs-cluster -p node11 -s node21 -N nodefile -r $(which ssh) -R $(which scp)
```

Use the **mmlscluster** command to view the cluster.

Apply IBM Storage Scale license

IBM Storage Scale requires a licensing designation before you can use it.

All IBM Storage Scale nodes require a license designation before they can be used. The FPO feature introduced a dedicated PFS license class **fpo**. In the IBM Storage Scale FPO cluster, all quorum and manager nodes require a server license. Based on the sample environment, **node11**, **node21**, and **node31** require a server license. The other nodes require an **fpo** license.

```
# mmchlicense server --accept -N node11,node21,node31

# mmchlicense fpo --accept -N
node12,node13,node14,node15,node16,node22,node23,node24,node25,node26,
node32,node33,node34,node35,node36
```

Use the **mmlslicense -L** command to view license information for the cluster.

Nodes with no disks in the file system are called as diskless nodes. Run the **mmchlicense client --accept -N** command to accept the client license for disks that have no disks in the IBM Storage Scale file system.

Start the IBM Storage Scale cluster to verify whether it starts successfully. Use the `mmstartup -a` command to start the IBM Storage Scale cluster and the `mmgetstate -a` command to view the state of the IBM Storage Scale cluster.

Create IBM Storage Scale Network Shared Disks (NSD)

To create the network shared disks (NSD) in IBM Storage Scale, create a disk file to be used as input to the `mmcrlnsd` command. The disk file defines the IBM Storage Scale pools and the NSDs. A recommended IBM Storage Scale pool configuration has two storage pools, a system pool for metadata only and a data pool.

- Storage Pools

- System pool contains all of the metadata disks and does not have FPO behavior enabled. The system pool should have a smaller block size than the data pool for performance reasons. If you choose to use **dataAndMetadata** disks in the system pool, you must set the system pool block size to be the same as the data pool block size as both the pools can have data. For the **dataAndMetadata** system pool, the block size 1M is recommended.
 - Data pool contains all of the data disks and has FPO behavior enabled by setting **allowWriteAffinity=yes**, **writeAffinityDepth=1**, and **blockGroupFactor=128**.

The chunk size can be calculated as **blockSize * blockGroupFactor**. Similar to the HDFS recommendation, the IBM Storage Scale FPO recommendation is **blockSize=2M * blockGroupFactor=64** for a chunk size of 128 MB

- NSD

- Every local disk to be used by IBM Storage Scale must have a matching entry in the disk stanza file
 - The device must match the device path of the disk.

Note: In the example, /dev/sda is not included because this is the OS disk.

If **MapReduce** intermediate and temporary data is stored on ext3/ext4 disks instead of IBM Storage Scale, make sure those disks are not included in the disk file or IBM Storage Scale will format them and include them in the IBM Storage Scale cluster.

- System pool disks:

- Should have **usage=metadataOnly**. It is possible to use **usage=dataAndMetadata** if there is a reason to have data on the system pool disks. The block size of the **dataAndMetadata** system pool must be the same as the block size of a data pool in the file system.
 - **failureGroup** must be a single number if **allowWriteAffinity** is not enabled (specify **allowWriteAffinity=no** for system pool definition when doing `mmcrlnsd` or `mmcrfs`) and it should be the same for all disks on the same node. If **allowWriteAffinity** is enabled for system pool, the failure group can be of format `rack,position,node`, for example, **2,0,1**; or, it can take the traditional single-number failure group format also.
 - Even when **allowWriteAffinity** is enabled for system pool, the metadata does not follow data locality rules; these rules apply only to data placement

- Data pool disks:

- Must have **usage=dataOnly**.
 - **failureGroup** must be of the format `[rack, position, node]`, where position is either 0 or 1 to represent top or bottom half of the rack. The sample environment does not have half racks, so the same position is used for all nodes. Especially, when position and node fields are ignored in the cluster, the failure group can be defined as a single number, `[rack, -, -]`.

Example of NSD disk file created by using the `mmcrlnsd` command:

```
%pool: pool=system blockSize=256K layoutMap=cluster allowWriteAffinity=no
%pool: pool=datapool blockSize=2M layoutMap=cluster allowWriteAffinity=yes writeAffinityDepth=1
blockGroupFactor=256

# gpfstest9
%nsd: nsd=node9_meta_sdb device=/dev/sdb servers=gpfstest9 usage=metadataOnly failureGroup=1 pool=system
```

```
%nsd: nsd=node9_data_sdf2 device=/dev/sdf servers=gpfstest9 usage=dataOnly failureGroup=1,0,1 pool=datapool
%nsd: nsd=node9_data_sdg2 device=/dev/sdg servers=gpfstest9 usage=dataOnly failureGroup=1,0,1 pool=datapool

#gpfstest10
%nsd: nsd=node10_meta_sda device=/dev/sda servers=gpfstest10 usage=metadataOnly failureGroup=2 pool=system
%nsd: nsd=node10_data_sde2 device=/dev/sde servers=gpfstest10 usage=dataOnly failureGroup=2,0,1 pool=datapool
%nsd: nsd=node10_data_sdg2 device=/dev/sdg servers=gpfstest10 usage=dataOnly failureGroup=2,0,1 pool=datapool

#gpfstest11
%nsd: nsd=node11_meta_sdb device=/dev/sdb servers=gpfstest11 usage=metadataOnly failureGroup=3 pool=system
%nsd: nsd=node11_data_sdf2 device=/dev/sdf servers=gpfstest11 usage=dataOnly failureGroup=3,0,1 pool=datapool
%nsd: nsd=node11_data_sdg2 device=/dev/sdg servers=gpfstest11 usage=dataOnly failureGroup=3,0,1 pool=datapool
```

If any disks are previously used by IBM Storage Scale, you must use the `-v no` flag to force IBM Storage Scale to use them again.

Note: Use the `-v no` flag only if you are sure that the disk can be used by IBM Storage Scale.

Use the `# mmcrnsd -F diskfile [-v no]` command to create NSDs and use the `mm1snsd -m` command to display the NSDs.

Apply IBM Storage Scale FPO configuration changes

IBM Storage Scale FPO requires several global IBM Storage Scale configuration changes to operate successfully.

Set the IBM Storage Scale page pool to 25% of system memory on each node. For Hadoop noSQL application, the page pool of IBM Storage Scale FPO can be configured for better performance, for example, 30% of physical memory.

In this example, all nodes have the same amount of memory, which is a best practice. If some nodes have different memory, set the page pool on a per-node basis by using the `-N` flag.

```
# TOTAL_MEM=$(cat /proc/meminfo | grep MemTotal | tr -d \'[:alpha:]\' | tr -d \'[:punct:]\' | tr -d \'[:blank:]\'")
# PAGE_POOL=$(( ${TOTAL_MEM}*25/(100*1024) ))
# mmchconfig pagepool=${PAGE_POOL}M
```

Start the IBM Storage Scale cluster:

```
# mmstartup -a
# mmgetstate -a
```

Use the `mm1sconfig` and `mmdiag` commands to see the configuration changes:

```
# mm1sconfig
# mmdiag --config
```

Create the IBM Storage Scale file system and pools

After you create NSDs, the IBM Storage Scale file system can be created.

To use FPO, a single file system is recommended. The following example creates a file system with mount point `/mnt/gpfs` that is set to auto mount. This mount point is used in Hadoop configuration later. The replication for both data and metadata is set to 3 replicas. Quotas are not activated on this file system. An inode size of 4096 is recommended for typical **MapReduce** data sizes `-S` and `-E` settings help improve performance for `mtime` and `atime` updates. The `mmcrfs` command also creates IBM Storage Scale storage pools based on the disk file %pools setting.

```
# mmcrfs gpfs-fpo-fs -F diskfile -T /mnt/gpfs -n 32 -m 3 -M 3 -r 3 -R 3 -i 4096 -A yes -Q no -S relatime -E no [-v no]
```

For more information on the pool configuration, see “[Create IBM Storage Scale Network Shared Disks \(NSD\)](#)” on page 689.

Mount the file system on all nodes:

```
# mmount all -a
```

Use the **mmlsfs** command to display the file system configuration:

```
# mmlsfs all
```

Use the **mmlsdisk** command to display the status of the NSDs:

```
# mmlsdisk gpfs-fpo-fs -L
```

Use the **mmdf** command to view the disk usage for the file system:

```
# mmdf gpfs-fpo-fs
```

Use the **mmlspool** command to view the storage pools:

```
# mmlspool gpfs-fpo-fs all -L
```

Create IBM Storage Scale Data Placement Policy

Before data can be written to the IBM Storage Scale file system that has more than one pool, you must apply a data placement policy.

In this example, all of the data goes to data pool.

```
# cat policyfile  
  
rule default SET POOL 'datapool'
```

After you create the rule file, use the **mmchpolicy** command to enable the policy:

```
# mmchpolicy gpfs-fpo-fs policyfile -I yes
```

Use the **mmlspolicy** command to display the currently active rule definition:

```
# mmlspolicy gpfs-fpo-fs -L
```

Create filesets for MapReduce intermediate and temporary data

To efficiently store MapReduce intermediate and temporary data, use filesets and policies to better emulate local disk behavior.

Note: If MapReduce intermediate and temporary data is not stored on IBM Storage Scale, **mapred.cluster.local.dir** in MRv1 or **yarn.nodemanager.log-dirs** and **yarn.nodemanager.local-dirs** in Hadoop Yarn does not point to the IBM Storage Scale directory, you do not need to go through this section.

Create an independent fileset

Consider using **--inode-space new [--inode-limit MaxNumInodes[:NumInodesToPreallocate]]** to create an independent fileset. This can improve the performance for the fileset but requires calculation for **MaxNumInodes** and **NumInodesToPreallocate**. **MaxNumInodes** must be eight times the number of files expected on the fileset, and

NumInodesToPreallocate must be half the value of **MaxNumInodes**. See the **mmcrfileset** man page to understand this option.

Use the **mmcrfileset** command to create two filesets, one for local intermediate data and one for temporary data:

```
# mmcrfileset gpfs-fpo-fs mapred-local-fileset  
  
# mmcrfileset gpfs-fpo-fs mapred-tmp-fileset
```

After the fileset is created, it must be linked to a directory under this IBM Storage Scale file system mount point. This example uses **/mnt/gpfs/mapred/local** for intermediate data and **/mnt/gpfs/tmp** for temporary data. As **/mnt/gpfs/mapred/local** is a nested directory, the directory structure must exist before linking the fileset. These two directories are required for configuring Hadoop.

```
# mkdir -p $(dirname /mnt/gpfs/mapred/local)  
  
# mmmlinkfileset gpfs-fpo-fs mapred-local-fileset -J /mnt/gpfs/mapred/local  
  
# mmmlinkfileset gpfs-fpo-fs mapred-tmp-fileset -J /mnt/gpfs/tmp
```

Use the **mmlsfileset** command to display fileset information:

```
# mmlsfileset gpfs-fpo-fs -L
```

The next step to setting up the filesets is to apply the IBM Storage Scale policy so the filesets act like local directories on each node. This policy instructs IBM Storage Scale not to replicate the data for these two filesets, and since these filesets are stored on the data pool, they can use FPO features that keeps local writes on local disks. Metadata must still be replicated three times, which can result in performance overhead. File placement policies are evaluated in the order they are entered, so ensure that the policies for the filesets appear before the default rule.

```
# cat policyfile  
  
rule 'R1' SET POOL 'datapool' REPLICATE (1,1) FOR FILESET ('mapred-local-fileset')  
  
rule 'R2' SET POOL 'datapool' REPLICATE (1,1) FOR FILESET ('mapred-tmp-fileset')  
  
rule default SET POOL 'datapool'  
  
# mmchpolicy gpfs-fpo-fs policyfile -I yes
```

Use the **mmlspolicy** command to display the currently active rule definition:

```
# mmlspolicy gpfs-fpo-fs -L
```

In each of these filesets, create a subdirectory for each node that run Hadoop jobs. Based on the sample environment, this script creates these subdirectories:

```
# cat mk_gpfs_local_dirs.sh  
  
#!/bin/sh for nodename in $(mmlsnode -N all); do  
  
mkdir -p /mnt/gpfs/tmp/${nodename}  
  
mkdir -p /mnt/gpfs/mapred/local/${nodename}  
  
done
```

After that, on `${nodename}` , `link /mnt/gpfs/tmp/${nodename} /hadoop/tmp; link /mnt/gpfs/mapred/local/${nodename} /hadoop/local`. Then, in Hadoop cluster, configure `/hadoop/tmp` as `hadoop.tmp.dir` in all Hadoop nodes; configure `/hadoop/local` as `mapred.cluster.local.dir` in MRv1 or `yarn.nodemanager.log-dirs` and `yarn.nodemanager.local-dirs` in Hadoop Yarn for Hadoop nodes.

To check that the rules are working properly, you can write some test files and verify their replication settings. For example:

Create some files:

```
# echo "test" > /mnt/gpfs/mapred/local/testRep1  
  
# echo "test" > /mnt/gpfs/testRep3
```

Use the `mmlsattr` command to check the replication settings

```
# mmlsattr /mnt/gpfs/mapred/local/testRep1  
  
replication factors  
  
metadata(max) data(max) file [flags]  
  
-----  
  
1 ( 3) 1 ( 3) /mnt/gpfs/mapred/local/testRep1  
  
# mmlsattr /mnt/gpfs/testRep3  
  
replication factors  
  
metadata(max) data(max) file [flags]  
  
-----  
  
3 ( 3) 3 ( 3) /mnt/gpfs/testRep3
```

Set file system permissions

Depending on how different users interact with IBM Storage Scale, you must create a user directory with permissions that allow users to create their own home directories.

```
# mkdir -p /mnt/gpfs/user  
  
# chmod 1777 /mnt/gpfs/user
```

To make sure that MapReduce jobs can write to the IBM Storage Scale file system, assign permissions to the CLUSTERADMIN user. CLUSTERADMIN is the user who starts Hadoop **namenode** and **datanode** service, for example, user hdfs.

```
# chown -R CLUSTERADMIN:CLUSTERADMINGROUP /mnt/gpfs  
  
# chmod -R +rx /mnt/gpfs
```

Use the `ls` command to verify the permission settings:

```
# ls -lR /mnt/gpfs
```

Basic Configuration Recommendations

Operating system configuration and tuning

Perform the following steps to configure and tune a Linux system:

1. deadline disk scheduler

Change all the disks defined to IBM Storage Scale to use the 'deadline' queue scheduler (cfg is the default for some distros, such as RHEL 6).

For each block device defined to IBM Storage Scale, run the following command to enable the deadline scheduler:

```
echo "deadline" > /sys/block/<diskname>/queue/scheduler
```

Changes made in this manner (echoing changes to sysfs) do not persist over reboots. To make these changes permanent, enable the changes in a script that runs on every boot or (generally preferred) create a udev rule.

The following sample script sets deadline scheduler for all disks in the cluster that are defined to IBM Storage Scale (this example must be run on the node with passwordless access to all the other nodes):

```
#!/bin/bash
/usr/lpp/mmfss/bin/mmllsnsd -X | /bin/awk ' { print $3 " " $5 } ' | \
/bin/grep dev |
while read device node ; do
device=$(echo $device | /bin/sed 's/\/dev\///' )
/usr/lpp/mmfss/bin/mmdsh -N $node "echo deadline >
/sys/block/$device/queue/scheduler"
Done
```

As previously stated, changes made by echoing to sysfs files (as per this example script) take effect immediately on running the script, but do not persist over reboots. One approach to making such changes permanent is to enable a udev rule, as per this example rule to force all block devices to use deadline scheduler after rebooting. To enable this rule, you can create the following file as /etc/udev/rules.d/99-hdd.rules):

```
ACTION=="add|change", SUBSYSTEM=="block", ATTR{device/model}=="*",
ATTR{queue/scheduler}="deadline"
```

In the next step, give an example of how to create udev rules that apply only to the devices used by IBM Storage Scale.

2. disk IO parameter change

To further tune the block devices used by IBM Storage Scale, run the following commands from the console on each node:

```
echo 16384 > /sys/block/<device>/queue/max_sectors_kb
echo 256 > /sys/block/<device>/queue/nr_requests
echo 32 > /sys/block/<device>/device/queue_depth
```

These block device tuning settings must be large enough for SAS/SATA disks. For /sys/block/<device>/queue/max_sectors_kb, the tuning value chosen must be less than or equal to /sys/block/<device>/queue/max_hw_sectors_kb. Many SAS/SATA devices allow setting 16384 for max_hw_sectors_kb, but not all devices may accept these values. If your device does not allow for the block device tuning recommendations above, try setting smaller values and cutting the recommendations in half until the tuning is successful. For example, if setting max_sectors_kb to 16384 results in a write error:

```
echo 16384 > /sys/block/sdd/queue/max_sectors_kb
-bash: echo: write error: Invalid argument
```

Try setting *max_sectors_kb* to 8192:

```
echo 8192 > /sys/block/sdd/queue/max_sectors_kb
```

If your disk is not SAS/SATA, check the disk specification from the disk vendor for tuning recommendations.

Note: If the *max_sectors_kb* of your disks is small (for example, 256 or 512) and you are not allowed to tune the above values (that is, you get an “invalid argument” as per the example above), then your disk performance might be impacted because IBM Storage Scale IO requests might be split into several smaller requests according to the limits *max_sectors_kb* places at the block device level.

As discussed in Step 1 tuning recommendations, any tuning done by echoing to sysfs files will be lost when a node reboots. To make such a tuning permanent, either create appropriate udev rules or place these commands in a boot file that is run on each reboot.

As udev rules are the preferred way of accomplishing this kind of block device tuning, give an example of a generic udev rule that enables the block device tuning recommended in steps 1 and 2 for all block devices. This rule can be enabled by creating the following rule as a file */etc/udev/rules.d/100-hdd.rules*:

```
ACTION=="add|change", SUBSYSTEM=="block", ATTR{device/model}=="*",
ATTR{queue/nr_requests}="256", ATTR{device/queue_depth}="32",
ATTR{queue/max_sectors_kb}="16384"
```

If it is not desirable to tune all block devices with the same settings, multiple rules can be created with specific tuning for the appropriate devices. To create such device specific rules, you can use the ‘KERNEL’ match key to limit which devices udev rules apply to (for example, KERNEL==*sdb*). The following example script can be used to create udev rules that tune only the block devices used by IBM Storage Scale:

```
#!/bin/bash
#clean up any existing /etc/udev/rules.d/99-hdd.rules files
/usr/lpp/mmfs/bin/mmdsh -N All "rm -f /etc/udev/rules.d/100-hdd.rules"
#collect all disks in use by GPFS and create udev rules one disk at a time
/usr/lpp/mmfs/bin/mmlsnsd -X | /bin/awk '{ print $3 " " $5 }' | \
/bin/grep dev |
while read device node ; do
device=$(echo $device | /bin/sed 's/\//\//')
echo $device $node
echo "ACTION==\"add|change\", SUBSYSTEM=\"block\", \
KERNEL==\"$device\", ATTR{device/model}==\"*\", \
ATTR{queue/nr_requests}=\"256\", \
ATTR{device/queue_depth}=\"32\", ATTR{queue/max_sectors_kb}=\"16384\" > \
/tmp/100-hdd.rules"
/usr/bin/scp /tmp/100-hdd.rules $node:/tmp/100-hdd.rules
/usr/lpp/mmfs/bin/mmdsh -N $node "cat /tmp/100-hdd.rules >> \
/etc/udev/rules.d/100-hdd.rules"
Done
```

Note: The previous example script must be run from a node that has ssh access to all nodes in the cluster. This previous example script will create udev rules that will set the recommended block device tuning on future reboots. To put the recommended tuning values from steps 1 and 2 in place immediately in effect, the following example script can be used:

```
#!/bin/bash
/usr/lpp/mmfs/bin/mmlsnsd -X | /bin/awk '{ print $3 " " $5 }' | \
/bin/grep dev |
while read device node ; do
device=$(echo $device | /bin/sed 's/\//\//')
/usr/lpp/mmfs/bin/mmdsh -N $node "echo deadline > \
/sys/block/$device/queue/scheduler"
/usr/lpp/mmfs/bin/mmdsh -N $node "echo 16384 > \
/sys/block/$device/queue/max_sectors_kb"
/usr/lpp/mmfs/bin/mmdsh -N $node "echo 256 > \
/sys/block/$device/queue/nr_requests"
/usr/lpp/mmfs/bin/mmdsh -N $node "echo 32 > \
/sys/block/$device/device/queue_depth"
Done
```

3. disk cache checking

On clusters that do not run Hadoop/Spark workloads, disks used by IBM Storage Scale must have physical disk write caching disabled, regardless of whether RAID adapters are used for these disks.

When running other (non-Hadoop/Spark) workloads, write caching on the RAID adapters can be enabled if the local RAID adapter cache is battery protected, but the write cache on the physical disks must not be enabled.

Check the specification for your RAID adapter to figure out how to turn on/off the RAID adapter write cache, as well as the physical disk write cache.

For common SAS/SATA disks without RAID adapter, run the following command to check whether the disk in question is enabled with physical disk write cache:

```
sdparm --long /dev/<diskname> | grep WCE
```

If WCE is 1, it means the disk write cache is on.

The following commands can be used to turn on/off physical disk write caching:

```
# turn on physical disk cache  
sdparm -S -s WCE=1 /dev/<diskname>  
# turn off physical disk cache  
sdparm -S -s WCE=0 /dev/<diskname>
```

Note: The physical disk read cache must be enabled no matter what kind of disk is used. For SAS/SATA disks without RAID adapters, run the following command to check whether the disk read cache is enabled or not:

```
sdparm --long /dev/<diskname> | grep RCD
```

If the value of RCD (Read Cache Disable) is 0, the physical disk read cache is enabled. On Linux, usually the physical disk read cache is enabled by default.

4. Tune vm.min_free_kbytes to avoid potential memory exhaustion problems.

When vm.min_free_kbytes is set to its default value, in some configurations it is possible to encounter memory exhaustion symptoms when free memory must be available. It is recommended that vm.min_free_kbytes be set to between 5~6 percent of the total amount of physical memory, but no more than 2GB should be allocated for this reserve memory.

To tune this value, add the following into /etc/sysctl.conf and then run 'sysctl -p' on Red Hat or SuSE:

```
vm.min_free_kbytes = <your-min-free-KBmemory>
```

5. OS network tuning

If your network adapter is 10Gb Ethernet adapter, you can put the following into /etc/sysctl.conf and then run **/sbin/sysctl -p /etc/sysctl.conf** on each node:

```
sunrpc.udp_slot_table_entries = 128  
sunrpc.tcp_slot_table_entries = 128  
net.core.rmem_max=4194304  
net.core.wmem_max=4194304  
net.core.rmem_default=4194304  
net.core.wmem_default=4194304  
net.core.netdev_max_backlog = 300000  
net.core.somaxconn = 10000  
net.ipv4.tcp_rmem = 4096 4224000 16777216  
net.ipv4.tcp_wmem = 4096 4224000 16777216  
net.core.optmem_max=4194304
```

If your cluster is based on InfiniBand adapters, see the guide from your InfiniBand adapter vendor.

If you bond two adapters and configure xmit_hash_policy=layer3+4 with bond mode 4 (802.3ad, the recommended bond mode), IBM Storage Scale of one node has only one TCP/IP connection with

another node in the cluster for data transfer. This might make the network traffic only over one physical connection if the network traffic is not heavy.

If your cluster size is not large (for example, only one physical switch is enough for your cluster nodes), you could try bonding mode 6 (balance-alb, no special support from switch). This might give better network bandwidth as compared with bonding mode 4 (802.3ad, require support from switch). See the [Linux bonding: 802.3ad \(LACP\) vs. balance-alb mode](#) link for advantages and disadvantages on Linux bonding 802.3ad versus balance-alb mode.

IBM Storage Scale configuration and tuning

Perform the following steps to tune the IBM Storage Scale cluster and file systems:

1. Data replica and metadata replica:

While creating IBM Storage Scale file systems, ensure that the replication settings meet the data protection needs of the cluster.

For production cluster over internal disks, it is recommended to take replica 3 for both data and metadata. If you have local RAID5 or RAID6 adapters with battery protected, you can take replica 2 for the data.

When a file system is created, the default number of copies of data and metadata are respectively defined by the -r (DefaultDataReplicas) and -m (DefaultMetadataReplicas) options to the **mmcrfs** command. Also, the value of -R (MaxDataReplicas) and -M (MaxMetadataReplicas) cannot be changed after the file system is created. Therefore, it is recommended to take 3 for -R/-M for flexibility to change the replica in the future.

Note: The first instance (copy) of the data is referred to as the first replica. For example, setting the DefaultDataReplicas=1 (by using -d 1 option to **mmcrfs**) results in only a single copy of each piece of data, which is typically not desirable for a shared-nothing environment.

Query the number of replicas that are kept for any specific file system by running the command:

```
/usr/lpp/mmfs/bin/mmfs mmqlsfs <filesystem_name> | egrep " -r| -m"
```

Change the level of data and metadata replication for any file system by running **mmchfs** by using the same -r (DefaultDataReplicas) and -m (DefaultMetadataReplicas) flags to change the default replication options and then **mmrestripefs** (with the -R flag) to restripe the file system to match the new default replication options.

For example,

```
/usr/lpp/mmfs/bin/mmchfs <filesystem_name> -r <NewDefaultDataReplicas> -m <NewDefaultMetadataReplicas>
/usr/lpp/mmfs/bin/mmrestripefs <filesystem_name> -R
```

2. Additional considerations for the file system:

When you create the file system, consider tuning the **/usr/lpp/mmfs/bin/mmcrfsparameters** file based on the characteristics of your applications:

-L

By default, the value is 4 MB for a file system log file. It is a good idea to create any file system with at least a 16 MB log file (-L 16M) or, if your application is sensitive to meta-operations, with at least a 32 MB log file (-L 32M).

-E

By default, the value is **yes**, which provides exact mtime. If your applications do not require exact mtime, you can change this value to **no** for better performance.

-S

The default value depends on the minimum release level of the cluster when the file system is created. If the minimum release level is 5.0.0 or greater, the default value is **relatime**. Otherwise, the default value is **no**, which causes the atime to be updated each time that the file

is read. If your application does not depend on exact atime, **yes** or **relatime** provides better performance.

--inode-limit

If you plan for the file system to contain many files, it is a good idea to set the value as large as possible to avoid getting errors that say "no inode space". You can estimate the value of this parameter with the following formula:

```
--inode-limit = (<metadata_disk_size> * <metadata_disk_number>)/(<inode_size> * DefaultMetadataReplicas)
```

For more information, see the topics *mmchfs command* and *mmcrfs command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

3. Define the data and the metadata distribution across the NSD server nodes in the cluster:

Ensure that clusters larger than four nodes are not defined with a single (dataAndMetadata) system storage pool.

For performance and RAS reasons, it is recommended that data and metadata are separated in some configurations (which means that not all the storage is defined to use a single dataAndMetadata system pool).

These guidelines focus on the RAS considerations that are related to the implications of losing metadata servers from the cluster. In IBM Storage Scale Shared Nothing configurations (which recommend setting the unmountOnDiskFail=meta option), a given file system is unmounted when the number of nodes experiencing metadata disk failures is equal to or greater than the value of the DefaultMetadataReplicas option defined for the file system (the -m option to the **mmcrfs** command as per above). So, for a file system with the typically configured value DefaultMetadataReplicas=3, the file system unmounts when metadata disks in three separate locality group IDs fail (when a node fails, all the internal disks in that node is marked down).

Note: All the disks in the same file system on a given node must have the same locality group ID.

The Locality ID refers to all three elements of the extended failure group topology vector (For example, the vector 2,1,3 could represent rack 2, rack position 1, node 3 in this portion of the rack). To avoid file system unmounts associated with losing too many nodes serving metadata, it is recommended that the number of metadata servers be limited when possible. Also metadata servers must be distributed evenly across the cluster to avoid the case of a single hardware failure (such as the loss of a frame/rack or network switch) leading to multiple metadata node failures.

Some suggestions for separation of data and metadata based on cluster size:

Total NSD Nodes in Cluster	Suggestions for Allocation of Data and Metadata Across NSDs
1-5	<p>All nodes must have both data and metadata disks.</p> <p>Depending on the available disks it is optional: both the data and metadata can be stored on each disk in this configuration (in which case, the NSDs are all defined as <i>dataAndMetadata</i>) or the disks can be specifically allocated for data or metadata.</p> <p>If the disk number per node is equal to or less than 3, define all the disks as <i>dataAndMetadata</i>.</p> <p>If the disk number per node is larger than 3, take the 1:3 ratio for <i>metadataOnly</i> disk and <i>dataOnly</i> disk if your applications are meta data IO sensitive. If your applications are not metadata IO sensitive, consider using 1 <i>metadataOnly</i> disk.</p>

Total NSD Nodes in Cluster	Suggestions for Allocation of Data and Metadata Across NSDs
6-9	<p>5 nodes must serve as metadata disks.</p> <p>Assign one node per virtual rack where each node is one unique failure group. Among these nodes, select 5 nodes with metadata disks and other nodes with data-only disks.</p> <p>For metadata disk number, if you are not considering IOPS for metadata disk, you can select one disk as metadata NSD from the above 5 nodes with metadata disks. Other disks from these 5 nodes are used for data disks. If considering IOPS for metadata disk, you could select 1:3 ratio for metadata:data.</p> <p>For example, if you have 8 nodes with 10 disks per node, you have totally 80 disks. However, if you are considering 1:3 ratio, you could have 20 disks for metadata and select 5 disks per node from the above 5 nodes as metadata NSD disks. All other disks are configured as data NSD.</p>
10 - 19	<p>There are several different layouts, for example, 2 nodes per virtual rack for 10-node cluster; for 20-node cluster, you can take every 4 nodes per virtual rack or every 2 nodes per virtual rack; for 15-node cluster, you can take every 3 nodes per virtual rack.</p> <p>You must keep at least 5 failure groups for meta data and data. This can ensure that you have enough failure groups for data stripe when you have failures from 2 failure groups.</p> <p>To make it simple, it is suggested that every 2 nodes are defined as a virtual rack, with the first element of the extended failure group kept the same for nodes in the same virtual rack, and every virtual rack must have a node with metadata disks defined.</p> <p>For example, for an 18-node cluster, node1~node18, node1 and node2 are considered as a virtual rack. You can select some disks from node1 as metadataOnly disks and other disks from node1 and all disks from node2 as dataOnly disks. Ensure that these nodes are in the same failure group (for example, all dataOnly disks from node1 are of failure group 1,0,1, all dataOnly disks from node2 are of failure group 1,0,2).</p>
20 or more	<p>Usually, it is recommended that the virtual rack number be greater than 4 but less than 32 with each rack of the same node number. Each rack is defined as one unique failure group and you have more than five failure groups that can tolerate failures from 2 failure groups for data stripe. Select one node from each rack to serve as meta data node.</p> <p>For example, for a 24-node cluster, you can split the clusters into 6 virtual racks with 4 nodes per rack. For 21-node cluster, it is recommended to take 7 virtual racks with 3 nodes per rack. For node number larger than 40, as a starting point, it's recommended that approximately every 10 nodes should be defined as a virtual rack, with the first element of the extended failure group kept the same for nodes in the same virtual rack. As for meta data, every virtual rack should have one node with metadataOnly disks defined. If you have more than 10+ racks, you could only select 5~10 virtual racks configured with metadata disks.</p> <p>As for how many disks must be configured as metadataOnly disks on the node which is selected for metadataOnly disks, this depends on the exact disk configuration and workloads. For example, if you configure one SSD per virtual rack, defining the SSD from each virtual rack as metadataOnly disks work well for most workloads.</p>

Note:

- a. If you are not considering the IOPS requirement from the meta IO operations, usually 5% of the total disk size in the file system must be kept for meta data. If you can predict how many files your

- file system is filled and the average file size, then the requirement of the meta space size could be calculated roughly.
- b. In a Shared Nothing framework, it is recommended that all nodes have similar disks in disk number and disk capacity size. If not, it might lead to hot disks when some nodes with small disk number or small disk capacity size are running out of disk space.
 - c. As for the number of nodes that are considered as one virtual rack, it is recommended to keep the node number even from each virtual rack.
 - d. It is always recommended to configure SSD or other fast disks as metadataOnly disks. This speeds up some maintenance operations, such as **mmrestripefs**, **mmdeldisk**, and **mmchdisk**.
 - e. If you are not sure about failure group definition, contact scale@us.ibm.com
4. When running a sharing nothing cluster, choose a failure group mapping scheme suited to IBM Storage Scale.
- Defining more than 32 failure groups IDs for a specific file system slows down the execution of a lot of concurrent disk space allocation operations, such as restripe operations **mmrestripefs -b**.
- On FPO-enabled clusters, defining more than 32 locality groups per failure group ID slows down the execution of restripe operations, such as **mmrestripefs -r**.
- To define an IBM Storage Scale FPO-enabled cluster containing a storage pool, set the option `allowWriteAffinity` to yes. This option can be checked by running the **mmfspool <fs-name> all -L** command. In FPO-enabled clusters, currently all disks on the same node must be assigned to the same locality group ID (three integer vector x,y,z), which also defines a failure group ID <x,y>. It is recommended that failure group IDs refer to sets of common resources, with nodes sharing a failure group ID having a common point of failure, such as a shared rack or a network switch.
5. Do not configure `allowWriteAffinity=yes` for a metadataOnly system pool.
- For a metadataOnly storage pool (not a dataAndMetadata pool), set `allowWriteAffinity` to no. Setting `allowWriteAffinity` to yes for metadataOnly storage pool slows down the inode allocation for the pool.
6. Any FPO-enabled storage pool (any pool with `allowWriteAffinity=yes` defined) must define `blockGroupFactor` to be larger than 1 (regardless of the value of `writeAffinityDepth`).
- When `allowWriteAffinity` is enabled, more RPC (Remote Procedure Call) activity might occur compared to the case of setting `allowWriteAffinity=no`.
- To reduce some of the RPC overhead associated with setting `allowWriteAffinity=1`, for pools with `allowWriteAffinity` enabled, it is recommended that the `BlockGroupFactor` be set to greater than 1. Starting point recommendations are `blockGroupFactor=2` (for general workloads), `blockGroupFactor=10` (for database workloads), and `blockGroupFactor=128` (Hadoop workloads).
7. Tune the block size for storage pools defined to IBM Storage Scale.

For storage pools containing both data and metadata (pools defined as `dataAndMetadata`), a block size of 1M is recommended.

For storage pools containing only data (pools defined as `dataOnly`), a block size of 2M is recommended.

For storage pools containing only metadata (pools defined as `metadataOnly`), a block size of 256K is recommended.

The following sample pool stanzas (used when creating NSDs via the **mmcrnsd** command) are based on the tuning suggestions from steps 4-7:

```
#for a metadata only system pool:
%pool: pool=system blockSize=256K layoutMap=cluster allowWriteAffinity=no
#for a data and metadata system pool:
%pool: pool=system blockSize=1M layoutMap=cluster allowWriteAffinity=yes
writeAffinityDepth=1 blockGroupFactor=2
#for a data only pool:
%pool: pool=datapool blockSize=2M layoutMap=cluster allowWriteAffinity=yes
writeAffinityDepth=1 blockGroupFactor=10
```

8. Tune the size of the IBM Storage Scale pagepool attribute by setting the pool size of each node to be between 10% and 25% of the real memory installed.

Note: The Linux buffer pool cache is not used for IBM Storage Scale file systems. The recommended size of the pagepool attribute depends on the workload and the expectations for improvements due to caching. A good starting point recommendation is somewhere between 10% and 25% of real memory. If machines with different amounts of memory are installed, use the -N option to **mmchconfig** to set different values according to the memory installed on the machines in the cluster. Though these are good starting points for performance recommendations, some customers use relatively small page pools, such as between 2-3% of real memory installed, particularly for machines with more than 256GB installed.

The following example shows how to set a page pool size equal to 10% of the memory (this assumes all the nodes have the same amount of memory installed):

```
TOTAL_MEM=$(cat /proc/meminfo | grep MemTotal | tr -d \'[:alpha:]\' | tr -d \'[:punct:]\' | tr -d \'[:blank:]\'')
PERCENT_OF_MEM=10
PAGE_POOL=$(( ${TOTAL_MEM} * ${PERCENT_OF_MEM} / (100 * 1024) ))
mmchconfig pagepool=${PAGE_POOL}M -i
```

9. Change the following IBM Storage Scale configuration options and then restart IBM Storage Scale.

Note: For IBM Storage Scale 4.2.0.3 or 4.2.1 and later, the restart of IBM Storage Scale can be delayed until the next step, because tuning **workerThreads** requires a restart.

Set each configuration option individually:

```
mmchconfig readReplicaPolicy=local
mmchconfig unmountOnDiskFail=meta
mmchconfig restripeOnDiskFailure=yes
mmchconfig nsdThreadsPerQueue=10
mmchconfig nsdMinWorkerThreads=48
mmchconfig prefetchaggressivenesswrite=0
mmchconfig prefetchaggressivenessread=2
```

For versions of IBM Storage Scale earlier than 5.0.2, also set one of the following values:

```
mmchconfig maxStatCache=512
mmchconfig maxStatCache=0
```

In versions of IBM Storage Scale earlier than 5.0.2, the stat cache is not effective on the Linux platform unless the Local Read-Only Cache (LROC) is configured. For more information, see the description of the **maxStatCache** parameter in the topic *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Set all the configuration options at once by using the **mmchconfig** command:

```
mmchconfig readReplicaPolicy=local,unmountOnDiskFail=meta,
restripeOnDiskFailure=yes,nsdThreadsPerQueue=10,nsdMinWorkerThreads=48,
prefetchaggressivenesswrite=0,prefetchaggressivenessread=2
```

For versions of IBM Storage Scale earlier than 5.0.2, also include one of the following expressions: **maxStatCache=512** or **maxStatCache=0**.

The **maxMBpS** tuning option must be set as per the network bandwidth available to IBM Storage Scale. If you are using one 10 Gbps link for the IBM Storage Scale network traffic, the default value of 2048 is appropriate. Otherwise scale the value of **maxMBpS** to be about twice the value of the network bandwidth available on a per node basis.

For example, for two bonded 10 Gbps links an appropriate setting for **maxMBpS** is:

```
mmchconfig maxMBpS=4000 # this example assumes a network bandwidth of about
2GB/s (or 2 bonded 10 Gbps links) available to Spectrum Scale
```

Note: In a special user scenario, such as active-to-active disaster recovery deployment, **restripeOnDiskFailure** must be configured as no for internal disk cluster.

Some of these configuration options do not take effect until IBM Storage Scale is restarted.

10. Depending on the level of code installed, follow the tuning recommendation for Case A or Case B:

- If running IBM Storage Scale 4.2.0 PTF3, 4.2.1, or any higher level, either set workerThreads to 512, or try setting workerThreads=8*cores per node (both require a restart of IBM Storage Scale to take effect). For lower code levels, setting worker1Threads to 72 (with the -i, immediate, option to **mmchconfig** does not require restarting IBM Storage Scale.)

```
mmchconfig workerThreads=512 # for Spectrum Scale 4.2.0 PTF3, 4.2.1, or  
any higher levels  
or  
mmchconfig workerThreads=8*CORES_PER_NODE # for Spectrum Scale 4.2.0 PTF3,  
4.2.1, or any higher levels
```

Change workerThreads to 512 (the default is 128) to enable additional thread tuning. This change requires that IBM Storage Scale be restarted to take effect.

Note: For IBM Storage Scale 4.2.0.3 or 4.2.1 or later, it is recommended that the following configuration parameters not be changed (setting workerThreads to 512, or (8*cores per node), auto-tunes these values): parallelWorkerThreads, logWrapThreads, logBufferCount, maxBackgroundDeletionThreads, maxBufferCleaners, maxFileCleaners, syncBackgroundThreads, syncWorkerThreads, sync1WorkerThreads, sync2WorkerThreads, maxInodeDeallocPrefetch, flushedDataTarget, flushedInodeTarget, maxAllocRegionsPerNode, maxGeneralThreads, worker3Threads, and prefetchThreads.

After you enable auto-tuning by tuning the value of workerThreads, if you previously changed any of these settings (parallelWorkerThreads, logWrapThreads, and so on) you must restore them back to their default values by running **mmchconfig <tunable>=Default**.

- For IBM Storage Scale 4.1.0.x, 4.1.1.x, 4.2.0.0, 4.2.0.1, 4.2.0.2, the default values work for most scenarios. Generally only worker1Threads tuning is required:

```
mmchconfig worker1Threads=72 -i # for Spectrum Scale 4.1.0.x, 4.1.1.x,  
4.2.0.0, 4.2.0.1, 4.2.0.2
```

For IBM Storage Scale 4.1.0.x, 4.1.1.x, 4.2.0.0, 4.2.0.1, 4.2.0.2, worker1Threads=72 is a good starting point (the default is 48), though larger values have been used in database environments and other configurations that have many disks present.

11. Customers running IBM Storage Scale 4.1.0, 4.1.1, and 4.2.0 must change the default configuration of trace to run in overwrite mode instead of blocking mode.

To avoid potential performance problems, customers running IBM Storage Scale 4.1.0, 4.1.1, and 4.2.0 must change the default IBM Storage Scale tracing mode from blocking mode to overwrite mode as follows:

```
/usr/lpp/mmfs/bin/mmtracectl --set --trace=def --tracedev-writemode=  
overwrite --tracedev-overwrite-buffer-size=500M # only for Spectrum  
Scale 4.1.0, 4.1.1, and 4.2.0
```

This assumes that 500MB can be made available on each node for IBM Storage Scale trace buffers. If 500MB are not available, then set a lower appropriately sized trace buffer.

12. Consider whether pipeline writing must be enabled.

By default, data ingestion node writes 2 or 3 replicas of the data to the target nodes over the network in parallel when pipeline writing is disabled (**enableRepWriteStream=0**). This takes additional network bandwidth. If pipeline writing is enabled, the data ingestion node only writes one replica over the network and the target node writes the additional replica. Enabling pipeline writing (**mmchconfig enableRepWriteStream=1** and restarting IBM Storage Scale daemon on all nodes) can increase IO write performance in the following two scenarios:

- Data is ingested from the IBM Storage Scale client and the network bandwidth from the data-ingesting client is limited.

- b. Data is written through rack-to-rack switch with limited bandwidth. For example, 30 nodes per rack, the bandwidth of rack-to-rack switch is 40Gb. When all the nodes are writing data over the rack-to-rack switch, each node gets only 40Gb/30, which is approximately 1.33Gb average network bandwidth.

For other scenarios, enableRepWriteStream must be kept as 0.

Optional IBM Storage Scale configuration and tuning

Note: A restart of IBM Storage Scale is needed to bring into effect any configuration option changes that do not successfully complete with the **-i** (immediate) option to **mmchconfig**. For example, changing the minMissedPingTimeout requires a restart.

The following configurations can be changed by using **mmchconfig** as per the needs of the system workload:

Configuration Options	Default	Recommended	Comment
metadataDiskWaitTimeForRecovery	2400(seconds)	Refer comment	Effective when <code>restripeOnDiskFailure</code> is set to yes. The default value is 40 minutes. This value must be long enough to cover the reboot time of the node with meta disk.
dataDiskWaitTimeForRecovery	3600(seconds)	Refer comment	Effective when <code>restripeOnDiskFailure</code> is set to yes. The default value is 60 minutes. This value must be long enough to cover the reboot time of the node with data disk.
syncBuffsPerIteration	100	100 (default)	It is recommended to not change the default value. Substantial improvements resulting from tuning this value have not been observed.

Configuration Options	Default	Recommended	Comment
minMissedPingTimeout	3(seconds)	10-60(seconds)	Sets the lower bound on a missed ping timeout. For FPO clusters, a longer grace time is desirable before marking a node as dead, as it impacts all associated disks. Additionally, when running MapReduce workloads, the CPU can become overly busy and cause delayed ping responses. However, a longer timeout implies delay in recovery. A value between 10– 60 seconds is recommended. This value generally provides a good balance between the time to detect the real failures and the rate of false failure detection triggered by a delayed ping response due to CPU or network overload.
leaseRecoveryWait	35	65	The default value is 35 and the recommended value is 65. Set a value lower than 65 if you want more rapid recovery from failures. Do not set a value lower than 35.
prefetchPct	20(% of pagepool parameter)	See comment	Used by IBM Storage Scale as a guideline to limit the page pool space used for prefetch or write-behind buffers. For MapReduce workloads, generally used for sequential read and write, increase this parameter up to its 60% of the maximum pagepool size.

Configuration Options	Default	Recommended	Comment
maxFilesToCache	4000	100000	Specifies the number of inodes to cache. Storing the inode of a file in cache permits faster re-access to the file while retrieving location information for data blocks. Increasing this number can improve throughput for workloads with high file reuse, as is the case with Hadoop MapReduce tasks. However, increasing this number excessively can cause paging at the file system manager node. The value must be large enough to handle the number of concurrently open files and allow caching of recently used files.
nsdInlineWriteMax	1,024	1,000,000 (or default tuning value)	Defines the amount of data sent in-line with write requests to the NSD server and helps to reduce overhead caused by internode communication.
nsdThreadsPerDisk	3	8	For NSDs that are each SATA/SAS disk, set 8 for file systems with the block size of 2MB and 16 for file systems with the block size of 1MB. For NSDs that are LUNs, each containing multiple physical disks, increase this accordingly to the number of drives per NSD.
nsdSmallThreadRatio	0	2	The ratio of the number of small threads to the number of large threads. The recommendation is to change this to 2 for most workloads.

Configuration and tuning of Hadoop workloads

All configuration options listed in this section are applicable only to Hadoop-like applications such as Hadoop and Spark:

Configuration	Default Value	Recommended	Comment
disableInodeUpdateOnFdatasync	No	Yes	

Configuration	Default Value	Recommended	Comment
dataDiskCacheProtectionMethod	0	2	Change this to 2 if you turn on dataOnly disk write cache (without battery protection).

Note: If the cluster is not dedicated for Hadoop workloads, take the default value for the above configurations.

For Hadoop-like workloads, one JVM process can open a lot of files. Therefore, tune the ulimit values:

```
vim /etc/security/limits.conf
# add the following lines at the end of /etc/security/limits.conf
* soft nofile 65536
* hard nofile 65536
* soft nproc 65536
* hard nproc 65536
```

kernel.pid_max

Usually, the default value is 32K. If you see the error allocate memory or unable to create new native thread, try to increase kernel.pid_max by adding kernel.pid_max=99999 at the end of /etc/sysctl.conf and then sysctl -p.

Configuration and tuning of database workloads

If the cluster is focused on database workloads such as SAP HANA/DB2 DPF/DashDB or DIO/AIO, the following configuration options must be tuned:

Configuration	Default Value	Recommended	Comment
enableLinuxReplicatedAo	N/A	Yes	The default value depends on the release of IBM Storage Scale.
preStealPct	1	See below	Only for Direct I/O.

Database workload customers using direct I/O must also enable the following preStealPct tuning depending on the IBM Storage Scale levels:

- 3.5 (any PTF level)
- 4.1.1 (below PTF 10)
- 4.2.0 (any PTF level)
- 4.2.1 (below PTF 2).

The database workload customers with direct I/O enabled who are running older code levels must tune preStealPct as follows:

```
echo 999 | mmchconfig preStealPct=0 -i
```

After upgrading to IBM Storage Scale from one of the previously referenced older code levels to a higher level (especially 4.1.1 PTF 10, 4.2.1 PTF 2, or 4.2.2.0 or higher), you can set the configuration option preStealPct=0 to its default value as follows:

```
echo 999 | mmchconfig preStealPct=1 -i
```

Configuring and tuning SparkWorkloads

1. Configure spark.shuffle.file.buffer.

By default, this must be configured on \$SPARK_HOME/conf/spark-defaults.conf.

To optimize Spark workloads on an IBM Storage Scale file system, the key tuning value to set is the `spark.shuffle.file.buffer` configuration option used by Spark (defined in a spark config file) which must be set to match the block size of the IBM Storage Scale file system being used.

The user can query the block size for an IBM Storage Scale file system by running: `mmlsfs <filesystem_name> -B`

The following is an example of tuning the `spark_shuffle_buffer_size` for a given file system:

```
spark_shuffle_file_buffer=$(/usr/lpp/mmf/bin/mmlsfs  
<filesystem_name> -B | tail -1 | awk '{ print $2 }')
```

Need to set the Spark configuration option `spark.shuffle.file.buffer` to the value assigned to `$spark_shuffle_file_buffer`.

Defining a large block size for IBM Storage Scale file systems used for spark shuffle operations can improve system performance. However, using a block size larger than 2M can offer useful improvements on typical hardware used in FPO configurations is not proven.

2. Configure `spark.local.dir` with local path.

Do not put the Spark's shuffle data into the IBM Storage Scale file system because this slows down the shuffle process.

Ingesting data into IBM Storage Scale clusters

MapReduce tasks perform best when input data is evenly distributed across cluster nodes. You can use the following approaches or a combination to ingest data for the first time and on an ongoing basis:

- Import data through a diskless IBM Storage Scale node. This ensures that the data is distributed evenly across all failure groups and all nodes within a failure group.
- If you have a large set of data to copy, it might help to use all cluster nodes to share ingest workload. Use a **write-affinity** depth of 0, along with as many cluster nodes with storage as possible to copy data in parallel.
- A **write-affinity** depth of 0 ensures that each node distributes data across as many nodes as possible. IBM Storage Scale policies can be used to enforce write-affinity depth settings based on fileset name, filename, or other attributes.
- Another mechanism to distribute data on ingest is to use **write-affinity depth failure** groups (WADFG) to control placement of the file replica. A WADFG setting of “*,*” ensures that all the file chunks are evenly distributed across all nodes. A placement policy can be used to selectively specify this attribute on the data set being ingested.

It is possible that even after employing the above techniques to ingest, the cluster might become unbalanced as nodes and disks are added or removed. You can check whether the data in the cluster is balanced by using the `mmdf` command. If data disks in different nodes are showing uneven disk usage, rebalance the cluster by running the `mmrestripefs -b` command. Keep in mind that the rebalancing command causes additional I/O activity in the cluster. Therefore, plan to run it at a time when workload is light.

Exporting data out of IBM Storage Scale clusters

In many applications, it might be required to export the output data into another system or application for further use. Hadoop native components such as Flume can be used for this purpose. If HDFS Transparency is used for Hadoop applications, the distcp feature is supported over IBM Storage Scale to export data into a remote HDFS file system or IBM Storage Scale file system. Additionally, since IBM Storage Scale provides POSIX semantics, custom scripts can be written to move data from an IBM Storage Scale cluster to any other POSIX-compliant file system. Consider using CNFS to copy the needed data directly.

If data must be exported into another IBM Storage Scale cluster, the AFM function can be used to replicate data into a remote IBM Storage Scale cluster.

Upgrading FPO

When the application that runs over the cluster can be stopped, you can shut down the entire GPFS cluster and upgrade FPO. However, if the application over the cluster cannot be stopped, you need to take the rolling-upgrade procedure to upgrade nodes.

During this kind of upgrade, the service is interrupted. In production cluster, service interrupt is not accepted by the customers. If such cases, you need to take the rolling upgrade to upgrade node by node (or failure group by failure group) while keeping GPFS service up in the other nodes.

The guide for rolling upgrade is as follows:

- Only upgrade nodes from the same failure group at the same time slot; not operate nodes from two or more failure groups because bringing nodes from more than 1 failure groups will make your data exposed in data lost risk.
- Not break the quorum relationship when bringing down the nodes from one failure group. Before you bring down the nodes in one failure group, you need to check the quorum node. If bringing the quorum node in the to-be-operated failure group will break the quorum relationship in the cluster, you need to exclude that node for the rolling upgrade of the failure group.

Prerequisites

- Ensure that all disks are in a ready status and up availability. You can check by issuing the `mmlsdisk -fs -name -L` command.
- Verify whether the upgraded-to GPFS version is compatible with the running version from [IBM Storage Scale FAQ in IBM Documentation](#). For example, you cannot upgrade GPFS from 3.4.0.x directly into 3.5.0.24. You need to upgrade to 3.5.0.0 first and then upgrade to the latest PTF. You also need to verify whether the operating system kernel version and the Linux distro version are compatible with GPFS from [IBM Storage Scale FAQ in IBM Documentation](#).
- Find a time period when the whole system work load is low or reserve a maintenance time window to do the upgrade. When cluster manager or file system manager is down intentionally or accidentally, another node is elected to take the management role. But it takes time to keep the cluster configuration and the file system data consistent.
- When a file system manager is elected by cluster manager, it does not change even if the file system is unmounted in this node. If the file system is mounted in other nodes, it is also ‘internal’ mounted in the file system manager. This does not affect your ability to unload the kernel modules and upgrade GPFS without a reboot.

Upgrade FPO as follows:

1. Disable auto recovery for disk failure

To do a rolling upgrade of GPFS, you must shut down GPFS in some nodes. Disks in those nodes are unreachable during that time. It is better to handle this disk down manually with the following step.

Run `mmchconfig stripeOnDiskFailure=no -i` in any node in cluster to disable auto recovery for disk failure. It is necessary to include `-i` option for this change to take effect immediately and permanently. GPFS `stripeOnDiskFailure` is a cluster-wide configuration. So you need to run it only once in any one node in your cluster.

2. Get the list of nodes for this upgrade cycle

Each upgrade cycle, you can only upgrade GPFS in nodes whose disks in it have the same first two numbers in failure group vector. Save node list in file `nodeList`, one node name per line in it. For general information on how to specify node names, see “[Specifying nodes as input to GPFS commands](#)” on page 201.

3. Get disk list in nodes for this upgrade cycle

Get all disks attached in nodes for this upgrade cycle. Save it in file `diskList`. Each line in file `diskList` saves a disk name. `mmlsdisk -L` command can show which disks belong to the nodes you want to upgrade in this cycle.

4. Stop applications by using GPFS file system

Confirming that there is no application which still opens file in GPFS file system. You can use **lsof** or **fuse** command to check whether there is still an open instance active for file in GPFS file system.

5. Unmount GPFS file system

Unmount GPFS file system in all nodes for this upgrade cycle through command:

```
mmumount <fsName> -N <nodeList>
```

Confirming file system has already unmounted in all related nodes through command:

```
mmllsmount <fsName> -L
```

6. Suspend disks

Suspend all attached disks in nodes for this upgrade cycle to make sure that GPFS does not try to allocate new data block from these disks. GPFS can still read valid data block from suspended disk

```
mmchdisk <fsName> suspend -d <diskList>
```

Confirming disks are suspended properly through command:

```
mmllsdisk <fsName>
```

7. Shut down GPFS

Shut down GPFS in all nodes for this upgrade cycle through command:

```
mmshutdown -N <nodeList>
```

Confirming GPFS is in down status in these nodes through command:

```
mmgetstate -a
```

8. Upgrade GPFS

You can upgrade GPFS packages in each node of this upgrade cycle.

For general information on how to install GPFS packages on nodes, see the following topics in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*:

- *Installing IBM Storage Scale on Linux nodes and deploying protocols*
- *Installing IBM Storage Scale on AIX nodes*
- *Installing IBM Storage Scale on Windows nodes*

9. Start GPFS

After all packages are ready, you can start up GPFS:

```
mmstartup -N <nodeList>
```

Confirming GPFS are in "Active" status in these nodes through command:

```
mmgetstate -a
```

10. Resume and start disks

When GPFS is in "Active" status in all nodes, you can resume disks which were suspended intentionally in Step 7.

```
mmchdisk <fsName> resume -a or
```

```
mmchdisk <fsName> resume -d <diskList>
```

When these disks are in "ready" status and if some of these disks are in "down" availability, you can start these disks through the following command:

```
mmchdisk <fsName> start -a or  
mmchdisk <fsName> start -d <diskList>
```

This might take a while since GPFS must do incremental data sync up to keep all data in these suspended disks are up to date. The time it needs depends on how much data has changed when the disks were kept in suspended status. You have to wait for **mmchdisk start** command to finish to do next step.

Confirming all disks are in ready status and up state through command:

```
mmlsdisk <fsName>
```

11. Mounts GPFS file system

When all disks in the file system are in up status you can mount file system:

```
mmmount <fsName> -N <nodeList>
```

Confirming GPFS file system is mounted properly through command:

```
mmlsmount <fsName> -L
```

Repeat Step 3 to Step 12 to upgrade GPFS in all nodes in your cluster.

12. Enable auto recovery for disk failure

Now you can enable auto recovery for disk failure through command:

```
mmchconfig restripeOnDiskFailure=no -i
```

It's necessary to includes **-i** option for this change to take effect immediately and permanently.

13. Upgrade GPFS cluster version and file system version

Issue **mmchconfig release=LATEST** and **mmchfs -V compat** to ensure the upgrade is successful and the cluster would never revert to the old build for minor GPFS upgrade. It is recommended to use **mmchfs -V compat** to enable backward-compatible format changes.

For major GPFS upgrade, consult IBM Support Center to verify compatibility between the different GPFS major versions, before issuing **mmchfs -V full**. For information about specific file system format and function changes when you upgrade to the current release, see [Chapter 27, “File system format changes between versions of IBM Storage Scale,” on page 267](#).

Monitoring and administering IBM Storage Scale FPO clusters

IBM Storage Scale supports the use of the simple network management protocol (SNMP) for monitoring the status and configuration of the IBM Storage Scale cluster. By using an SNMP application, the system administrator can get a detailed view of the system and receive instant notification of important events, such as a node or disk failure.

The SNMP agent software consists of a master agent and a set of subagents, which communicate with the master agent through an agent/subagent protocol, the AgentX protocol in this case.

The SNMP subagent runs on a collector node of the IBM Storage Scale cluster. The collector node is designated by the system administrator by using the **mmchnode** command.

The Net-SNMP master agent, also called as the SNMP daemon, or **snmpd**, must be installed on the collector node to communicate with the IBM Storage Scale subagent and with your SNMP management application. Net-SNMP is included in most Linux distributions and must be supported by your Linux vendor.

For more information about enabling SNMP support, see the *GPFS SNMP support* topic in the *IBM Storage Scale: Problem Determination Guide*.

Refer to the GPFS SNMP support topic in the *IBM Storage Scale: Administration Guide* for further information about enabling SNMP support.

If using IBM BigInsights®

When you install IBM Storage Scale, you can enable IBM Storage Scale monitoring using the IBM BigInsights installation program. If the monitoring was not enabled at the time of installation, it can be done later by installing the Net-SNMP master agent on the collector node to communicate with the IBM Storage Scale subagent and the IBM BigInsights Console. Detailed instructions are provided in the Enabling monitoring for GPFS topic in the IBM InfoSphere® BigInsights Version 2.1.2 documentation.

If using Cloudera HDP

Once the IBM Storage Scale service is installed, you can enable the IBM Storage Scale management GUI through Ambari. For more information, see the *IBM Storage Scale management GUI* topic under the big data and analytics support guide.

If using Platform Symphony® or Hadoop distribution from other vendor

You can leverage IBM Storage Scale SNMP integration for centralized monitoring of the IBM Storage Scale cluster. Follow the procedure outlined in the *GPFS SNMP support* topic in the *IBM Storage Scale: Problem Determination Guide*.

Rolling upgrades

During a regular upgrade, the IBM Storage Scale service is interrupted. For a regular upgrade, you must shut down the cluster and suspend the application workload of the cluster. During a rolling upgrade, there is no interruption in the IBM Storage Scale service. In a rolling upgrade, the system is upgraded node by node or failure group by failure group. During the upgrade, IBM Storage Scale runs on a subset of nodes.

In a rolling upgrade, nodes from the same failure group must be upgraded at the same time. If nodes from two or more failure groups stop functioning, only a single data copy is available online. Also, if the quorum node stops functioning, the quorum relationship in the cluster is broken. Therefore, the quorum node must be excluded from the rolling upgrade of the failure node.

Performing a rolling upgrade

This topic lists the steps to perform a rolling upgrade.

- Ensure that the status of all disks is Ready and the availability is Up by running the **mmldisk <fs-name> -L** command.
- Verify the compatibility of the new IBM Storage Scale version with the running version by reviewing the [IBM Storage Scale FAQ in IBM Documentation](#). For example, IBM Storage Scale cannot be upgraded from 3.4.0.x to 3.5.0.24 before being upgraded to 3.5.0.0.
- Verify the compatibility of the planned upgrade system kernel and Linux distro versions with IBM Storage Scale by reviewing the [IBM Storage Scale FAQ in IBM Documentation](#).
- While performing maintenance on the cluster manager and the file system manager nodes, the nodes fail over automatically. However, you must manually assign the cluster manager and the file system manager to other nodes by using the **mmchmgr** command when the cluster is not busy.

1. Disable auto recovery for disk failure.

To upgrade a node, shut down IBM Storage Scale running on the node. When IBM Storage Scale is shut down, disks in the node cannot be reached. Instead of letting the disks fail and the automatic recovery initiate, temporarily disable auto recovery.

Run the **mmchconfig restripeOnDiskFailure=no -i** command to disable auto recovery for disk failure. With the **-i** option, the parameter takes effect immediately and permanently. For example, in small clusters, the node number is less than 30 nodes. Therefore, it takes a shorter time for IBM Storage Scale to synchronize the configuration. For large clusters, the node number is in hundreds. Therefore, the time taken to synchronize the configuration is longer. The **restripeOnDiskFailure** parameter is a cluster-wide configuration.

After disabling auto recovery, check for auto recovery in the file system manager by running the following commands:

- If there are multiple file systems in the cluster, run **mm1smgr** command to check the fs manager of a single file system.
- Log in to the fs manager of the file system and run **ps -elf | grep -e tschdisk -e tsrestripes** command. If there are processes running, wait for them to complete.

2. Select the nodes that must be upgraded and schedule the time of each upgrade.

In each upgrade cycle, you can only upgrade IBM Storage Scale on nodes where the disks have the same first two numbers in the failure group. Save the list of nodes in the nodeList file with one node name on each line. Save a list of the disks on the nodes that will be upgraded in this cycle in the diskList file, with each line containing an NSD name. Run the **mm1sdisk Device -M** command to check which disks belongs to which node.

3. Stop all applications that are using the IBM Storage Scale file system before stopping IBM Storage Scale. To check for open files in the file system, run the **lsof** or the **fuse** command.

4. Unmount the IBM Storage Scale file system on all nodes by running the following command:

mmumount <fsName> -N <nodeList>

To confirm that the file system has been unmounted on all related nodes, run the following command:
mm1smount <fsName> -L

5. Suspend all disks in the nodes so that IBM Storage Scale does not allocate new data blocks from these disks. IBM Storage Scale can still read data block from suspended disks by running the following command:

mmchdisk <fsName> suspend -d <diskList>

To confirm that all disks are suspended properly, run the following command: **mm1sdisk <fsName>**

6. Shut down IBM Storage Scale on the nodes by running the following command:

mmshutdown -N <nodeList>

To confirm IBM Storage Scale has stopped functioning on these nodes, run the following command:
mmgetstate -a

Upgrade IBM Storage Scale packages on each node. For information on how to install IBM Storage Scale packages on node, see the following topics:

- *Installing IBM Storage Scale on Linux nodes and deploying protocols in IBM Storage Scale: Concepts, Planning, and Installation Guide*
- *Installing IBM Storage Scale on AIX nodes in IBM Storage Scale: Concepts, Planning, and Installation Guide*
- *Installing IBM Storage Scale on Windows nodes in IBM Storage Scale: Concepts, Planning, and Installation Guide*

After everything has been installed and the portability layer has been built, start IBM Storage Scale by running the following command: **mmstartup -N <nodeList>**

To confirm that IBM Storage Scale is active on the upgraded nodes, run the following command:
mmgetstate -a.

Resume all the suspended disks by running the following commands: **mmchdisk <fsName> resume -a** or **mmchdisk <fsName> resume -d <diskList>**.

If some of the suspended disks are in the Down availability, start these disks by running the following command: **mmchdisk <fsName> start -a** or **mmchdisk <fsName> start -d <diskList>**.

This may take a while because IBM Storage Scale is performing an incremental data sync up to keep the data in these suspended disks up-to-date. The time taken depends on the data that has been changed while the disks were kept in the Suspended status. Wait for the **mmchdisk <fsName> start [. . .]** command to finish before moving on to the next step.

To confirm that all disks are in the ready state, run the following command: **mm1sdisk <fsName>**.

7. When all the disks in the file system are functioning, mount the file system by running the following command: **mmmount <fsName> -N <nodeList>**

Confirm that the IBM Storage Scale file system has mounted by running the following command: **mmlsmount <fsName> -L**

8. Perform Step through Step to upgrade IBM Storage Scale on all nodes in the cluster.

9. To enable auto recovery for disk failure, run the following command: **mmchconfig restripeOnDiskFailure=yes -i**

Ensure that you use the **-i** option so that this change takes effect immediately and permanently.

10. Upgrade the IBM Storage Scale cluster version and file system version

If all applications run without any issues, run the **mmchconfig release=LATEST** command to upgrade the cluster version to the latest. Then, run the **mmchfs -V compat** command to ensure that the upgrade is successful. To enable backward-compatible format changes, run **mmchfs -V compat**.

Note: After running the **mmchconfig release=LATEST** command, you cannot revert the cluster release version to an older version. After running the **mmchfs -V compat** command, you cannot revert the file system version to an older version.

For major IBM Storage Scale upgrade, check [IBM Storage Scale FAQ in IBM Documentation](#) or contact scale@us.ibm.com before running the **mmchfs -V full** command to verify the compatibility between the different IBM Storage Scale major versions. For information about specific file system format and function changes, see Chapter 27, “File system format changes between versions of IBM Storage Scale,” on page 267.

Upgrading other infrastructure

The same process of choosing nodes should be used when upgrading hardware firmware, operation system kernel or other components that require you to take IBM Storage Scale down on the node.

The IBM Storage Scale FPO cluster

When a node reboots due to hardware or software issues, IBM Storage Scale can be started and the file system can be mounted if autoload is configured as yes in the mmchconfig command. In a typical FPO deployment, each node has several local attached disks. When a node stops functioning, disks attached to the node are made unavailable from the cluster.

Note: Do not reboot a node if the file system is still mounted.

After the node is rebooted, the disk status of the node is uncertain. The status of the node is dependent upon the auto recovery configuration (**mmlsconfig restripeOnDiskFailure**) and the IO operations over the cluster.

Restarting a large IBM Storage Scale cluster

A cluster might have to be restarted because of an OS upgrade. On large FPO clusters auto-recovery must be disabled before restarting IBM Storage Scale.

- Ensure that the status of all disks is Ready and the availability is Up by running the **mmlsdisk <fs-name> -L** command.
- Verify the compatibility of the planned upgrade system kernel and Linux distro versions with IBM Storage Scale by reviewing the [IBM Storage Scale FAQ in IBM Documentation](#).

1. Disable auto recovery for disk failure.

When IBM Storage Scale stops functioning, some nodes might not shut down. This might bring some disks down from the fast nodes and might trigger auto recovery. To avoid this, temporarily disable auto recovery.

Run the **mmchconfig restripeOnDiskFailure=no -i** command to disable auto recovery for disk failure. With the **-i** option, the parameter takes effect immediately and permanently.

For example, in small clusters, the node number is less than 30 nodes. Therefore, it takes a shorter time for IBM Storage Scale to synchronize the configuration. For large clusters, the node number is in hundreds. Therefore, the time taken to synchronize the configuration is longer. The **restripeOnDiskFailure** parameter is a cluster-wide configuration.

After disabling auto recovery, check for auto recovery in the file system manager by running the following commands:

- If there are multiple file systems in the cluster, run **mmlsmgr** command to check the fs manager of a single file system.
- Log in to the fs manager of the file system and run **ps -elf | grep -e tschdisk -e tsrestripefs** command. If there are processes running, wait for them to complete.

2. Stop all applications that are using the IBM Storage Scale file system. To check for open files in the file system, run the **lsof** or the **fuse** command.

For example, to check if IBM Storage Scale file system has processes using it run the following commands: **lsof +f -- /dev/name_of_SpectrumScale_filesystem** or **fuser -m /mount_point_of_SpectrumScale_filesystem**

3. Unmount the IBM Storage Scale file system on all nodes for this upgrade cycle by running the following command:

mmumount <fsName> -a

To confirm that the file system has been unmounted on all related nodes, run the following command: **mmlsmount <fsName> -L**

4. To disable the Automatic mount option, run the following command: **mmchfs <fsName> -A no**

Note: In a large cluster, some nodes might take a while to start. If the -A option is not set to **no**, unnecessary disk IO might cause some disks from slow nodes to be marked as non functional.

5. Shut down IBM Storage Scale on the nodes by running the following command:

mmshutdown -N <nodeList>

To confirm IBM Storage Scale has stopped functioning on these nodes, run the following command: **mmgetstate -a**

6. Upgrade IBM Storage Scale or perform the maintenance procedure on the whole cluster.

7. Start IBM Storage Scale cluster.

After everything has been installed and the portability layer has been built, start IBM Storage Scale by running the following command: **mmgetstate -a**.

To confirm that IBM Storage Scale is active on the upgraded nodes, run the following command: **mmgetstate -a**.

8. When IBM Storage Scale is active on all nodes, check the state of all disks by running the following command: **mmlsdisk <fsName> -e**.

If some disks in the file system do not have the Up availability and the Ready status, run the **mmchdisk <fsName> start -a** command so that the disks start functioning. Run the **mmchdisk <fsName> resume -a** command so that the suspended and to-be-emptied disks become available.

9. When all the disks in the file system are functioning, mount the file system by running the following command: **mmmount <fsName> -N <nodeList>**

Confirm that the IBM Storage Scale file system has mounted by running the following command: **mmlsmount <fsName> -L**

10. To enable auto recovery for disk failure, run the following command: **mmchconfig restripeOnDiskFailure=yes -i**

Ensure that you use the -i option so that this change takes effect immediately and permanently.

11. To enable the Automatic mount option, run the following command: **mmchfs <fsName> -A yes**.

12. If you have upgraded IBM Storage Scale version in step 6, upgrade the IBM Storage Scale cluster version and file system version.

If all applications run without any issues, run the `mmchconfig release=LATEST` command to upgrade the cluster version to the latest. Then, run the `mmchfs -V compat` command to ensure that the upgrade is successful. To enable backward-compatible format changes, run `mmchfs -V compat`.

Note: After running the `mmchconfig release=LATEST` command, you cannot revert the cluster release version to an older version. After running the `mmchfs -V compat` command, you cannot revert the file system version to an older version.

For major IBM Storage Scale upgrade, check [IBM Storage Scale FAQ in IBM Documentation](#) or contact scale@us.ibm.com before running the `mmchfs -V full` command to verify the compatibility between the different IBM Storage Scale major versions. For information about specific file system format and function changes, see [Chapter 27, “File system format changes between versions of IBM Storage Scale,” on page 267](#).

Failure detection

The node state

Learn how to find the state of the nodes in an IBM Storage Scale cluster.

To check the node state, issue the `mmgetstate` command with or without the `-a` option, as in the following examples:

- 1) `mmgetstate`
- 2) `mmgetstate -a`

Be aware of the differences between the **down**, **unknown**, and **unresponsive** states:

- A node in the **down** state is reachable but the GPFS daemon on the node is not running or is recovering from an internal error.
- A node in the **unknown** state cannot be reached from the node on which the `mmgetstate` command was run.
- A node in the **unresponsive** state is reachable but the GPFS daemon on the node is not responding.

To follow up on investigating the state of a node, check if the node is functioning or has a network issue.

For more information, see the topic *mmgetstate command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

The disk state

This topic describes how to check the disk state in the IBM Storage Scale cluster.

To check the state of the disks in the IBM Storage Scale cluster, run `mmlsdisk -e` command. This command lists all the disks that do not have the Available or Up status.

IBM Storage Scale log

This topic describes the IBM Storage Scale log files.

The IBM Storage Scale log files are saved in the `/var/adm/ras/` directory on each node. Each time the IBM Storage Scale daemon starts, a new log file is created. The `mmfs.log.latest` log file is the link to the latest log. On Linux, all additional information is sent to the system log in `/var/log/messages`.

Because the IBM Storage Scale cluster manager and file system managers handle cluster issues such as node leaves or disk down events, monitor the IBM Storage Scale log on the cluster manager and file system manager to get the best view of the cluster and file system status.

Disk Failures

This section describes how to handle a disk failure.

In an FPO deployment model with IBM Storage Scale the **restripeOnDiskFailure=yes** configuration parameter should be set to yes. When a disk is not functioning, auto recovery enables the disk to start functioning. Auto recovery enlists the help of any node in the cluster to help recover data. This may affect the file system I/O performance on all nodes, because data might have to be copied from a valid disk to recover the disk.

Stopping the disk failure auto recovery operation

The auto recovery operation can impact the I/O performance across the cluster. To avoid this problem, you can stop auto recovery manually and restart it later when the cluster is not so busy. The disks that are not functioning must be recovered to protect your data.

Run the **mmlsdisk -e** command to see the disks that do not have the Up availability and the Ready status. If all the disks in the file system are functioning correctly, the system displays the following message: 6027-623 All disks up and ready.

1. To stop the auto recovery process, stop the **tschdisk** and **tsrestripes** processes on the file system manager node. Log in to the IBM Storage Scale file system manager node. Retrieve the **tschdisk** and **tsrestripes** command processor ID through the **ps -elf | grep -e tschdisk -e tsrestripes** command.

Alternatively, check the IBM Storage Scale log (/var/adm/ras/mmfs.log.latest) in the file system manager node to see whether a **tschdisk** command is still running. When the **restripefs** command is invoked by the auto recovery and is still running, the command log message is redirected to /var/adm/ras/restripefsOnDiskFailure.log.<timestamp>(IBM Storage Scale 4.1 and IBM Storage Scale 4.1.1) or /var/adm/ras/autorecovery.log.<timestamp>(IBM Storage Scale 4.1.1 PTF1 and later).

2. Take the following steps to stop the **tschdisk** and **tsrestripes** command processes:

- a. Make a list of the file system manager nodes in the cluster. The list must include the file system manager node of each file system that is affected. To list the file system manager nodes, go to a file system manager node and issue the following command:

```
mmlsmgr
```

This command is in the directory /usr/lpp/mmfs/bin.

- b. Do the following actions for the **tschdisk** and **tsrestripes** processes on each of the file system manager nodes in your list:

i) If you are not connected to a file system manager node, connect to it with ssh.

ii) Issue the following command to list the back-end processes that are running and their command IDs:

```
mmfsadm command list all
```

In the following example, the **tsrestripes** process is running in the back end (line 6) and its command ID is #92 (line 5):

```
# mmfsadm command list all
CrHashTable 0x7F7E64001A08 n 4
cmd sock 75 cookie 3489916426 owner 12912 id 0x2D7ADC0785000064(#100) uses 1 type 14 start
1531294737.470181
flags 0x106 SG none line 'command list all'
cmd sock 70 cookie 2102087586 owner 4450 id 0x2D7ADC078500005C(#92) uses 1 type 13 start
1531294660.218091
flags 0x117 SG fpofs line 'tsrestripes /dev/fpofs -r'
hold PIT/repair waitTime 6.082489
```

- iii) If a back-end process is running, issue the following command to stop it:

```
mmfsadm command stop <commandID>
```

where <commandID> is the command ID of the back-end process from the previous step. The following example uses command ID 92 from the example in the previous step:

```
mmfsadm command stop 92
```

iv) Run the **mmfsadm** command again to verify that the process is no longer running:

```
mmfsadm command list all
```

Starting the disk failure recovery

This topic lists the steps to start the disk failure recovery.

1. To check the disks that are not in the Ready state, run the **mmlsdisk -e** command.
2. Resume the disks that have the Suspended status.

If there are multiple suspended disks, create a file that lists all the suspended disks one nsd name per line before you resume the disks. Resume the suspended disks by issuing the following command:

```
mmchdisk <fsName> resume -d <suspendDisk>List>
```

Check the disk status again by running the **mmlsdisk -e** command and confirm that all disks are now in the Ready state. If some disks are still in the Suspended state, there might be a hardware media or a connection problem. Save the names of these disks in the brokenDiskList file.

3. Save the disks that do not have the Up availability in the downDiskList file. Each line in downDiskList file stores a disk name. Start these disks by running the following command:

```
mmchdisk <fsName> start -d <downDiskList>
```

Check the disk status by running the **mmlsdisk -e** command to confirm that all disks have the Up availability. Disks that do not have the Up availability might have a hardware media or connection problem. Save the names of these disks in the tobeSuspendDiskList file. Suspend these disks by running the following command:

```
mmchdisk <fsName> suspend -d <tobeSuspendDiskList>
```

4. If a disk is in Suspended status after you restart it, there might be a hardware media or connection problems. To keep your data safe, migrate it to the suspended disks by running the following command:

```
mmrestripefs <fsName> -r
```

After a file system stripe, all data in the suspended disks is migrated to other disks. At this point, all the data in the file system has the desired level of protection.

5. Check the disk connections and the disk media for disks that are in the Suspended state and repeat step 2 through step 4. If a failure occurs again, delete these disks from file system by running the **mmdeledisk** command.

For example,

```
mmdeledisk <fs-name> "broken-disk1;broken-disk2"
```

If one file has some replica on down disks and if you make an update against this replica on down disks, the inode gets marked with the dataupdatemiss or metaupdatemiss flags (if the replica on down disks is metadata, it will be metaupdatemiss; if the replica on down disks is data, it will be dataupdatemiss). You could run **mmlsattr -d -L /path/to/file** to check these flags. These two flags could only be cleaned by **mmchdisk Device start**. If some down disks cannot be brought back with **mmchdisk Device start**, these flags will be kept even if you run **mmdeledisk**.

or **mmrestripefs -r**. To remove these flags, you could stop one NSD disk and then run **mmchdisk start** to bring it back immediately. This will clean up all the missupdate flags.

If you are unable to delete a broken disk, contact IBM support.

Handling physically broken disk

If a disk is physically broken, it cannot be recovered by auto recovery or manual recovery. Do not keep broken disks in the file system and schedule to delete them from the file system.

Deleting disks when auto recovery is not enabled (check this by **mmlsconfig restripeOnDiskFailure**):

Deleting NSD disks from the file system can trigger disk or network traffic because of data protection. If your cluster is busy with application IO and the application IO performance is important, schedule a maintenance window to delete these broken disks from your file system. Follow the steps in the “[Starting the disk failure recovery](#)” on page 717 section to check if a disk is physically broken and handle the broken disks.

Deleting disks when auto recovery is enabled (check this by **mmlsconfig restripeOnDiskFailure**):

When the IO operation is being performed on the physically broken disks, IBM Storage Scale marks the disks as non-functional. Auto recovery suspends the disks if it fails to change the availability of the disk to Up and restripes the data off the suspended disks. If you are using IBM Storage Scale 4.1.0.4 or earlier, deleting the non-functional disks triggers heavy IO traffic (especially for metadata disks). On IBM Storage Scale 4.1.0.4, **mmdeldisk** command has been improved. If the data on non-functional disks have been restriped, the disk status will be Emptied. The **mmdeldisk** command deletes the non-functional disks with the Emptied status without involving additional IO traffic.

Node failure

In an FPO deployment, each node has locally attached disks. When a node fails or has a connection problem with other nodes in a cluster, disks in this node become unavailable. Reboot a node to repair a hardware issue or patch the operating system kernel. Both these cases are node failures.

If auto recovery is enabled, that is, the **restripeOnDiskFailure=yes** parameter is set to yes, and a failed node is recovered within auto recovery wait time (check the details described in the [Auto Recovery for Disk Failure](#) section), auto recovery handles the node failure automatically by bringing up down disks and ensuring all data has the desired replication. If a node is not recovered within the auto recovery wait time, auto recovery migrates the data off the disks in the failed node to other disks in cluster.

Reboot node intentionally

Automatic recovery of a node

If you want to reboot a node or enable some configuration change that requires a reboot and have it recovered without auto recovery, check the auto recovery wait time. The auto recovery wait time is defined by the minimum value of minDiskWaitTimeForRecovery, metadataDiskWaitTimeForRecovery and dataDiskWaitTimeForRecovery. By default, minDiskWaitTimeForRecovery is 1800 seconds, metadataDiskWaitTimeForRecovery is 2400 seconds and dataDiskWaitTimeForRecovery is 3600 seconds. If the reboot is completed within the auto recovery wait time, it is safe to unmount the file system, shut down IBM Storage Scale, and reboot your node without having to disable auto recovery.

Manual recovery of a node

When you want to perform hardware maintenance for a node that must be shut down for a long time, follow the same steps mentioned in [IBM Storage Scale Rolling Upgrade Procedure](#) and perform hardware maintenance.

Node crash and boot up

This topic lists the steps to recover a node automatically or manually.

Recovering a node automatically

When a node crashes due to kernel or other critical issues and is recovered within the auto recovery wait period, IBM Storage Scale cluster manager can add this node automatically and IBM Storage Scale auto recovery brings up disks and repairs the dirty data in disks attached in the node. Check if the node and the disks in the node work normally and the data in the disks is updated.

1. Check whether IBM Storage Scale state is active on all nodes in the cluster by running the **mmgetstate -a** command. If a node is functional but IBM Storage Scale is not active, check IBM Storage Scale log (/var/adm/ras/mmfs.log.latest) on the node and run the **mmstartup** command after the issue in the log has been resolved.
2. Check if any disk does not have the Up availability and the Ready state by running the **mmlsdisk -e** command. If all disks in the file system are in the Ready state, the system displays the following message 6027-623 All disks up and ready. Perform the disk recovery operation to change the state of all disks to Ready.
3. Run the **mmlsdisk** command to confirm that there is no warning message at the end of output.

If you see the following message, there are data replicas on suspended and to-be-emptied disks.

If the suspended and to-be-emptied disks are not physically broken, recover them and run the **mmrestripefs -r** command to fix the warning message. If the disks are physically broken, suspend them and run the **mmrestripefs -r** command to fix the warning message.



Attention: Due to an earlier configuration change the file system may contain data that is at risk of being lost.

Recovering a node manually

When a node crashes and is not recovered while auto recovery is temporarily suspended, start the node and recover the disks. In this case, IBM Storage Scale auto recovery migrates all data from disks in this node to other valid disks in cluster.

1. Find the root cause of the node crash, fix it, and recover the node.
2. If IBM Storage Scale autoload configuration is disabled, start the IBM Storage Scale daemon by running the **mmstartup** command. Check if the node state is Active. If the state is not active after a few minutes, check IBM Storage Scale log (/var/adm/ras/mmfs.log.latest) on this node and run the **mmstartup** command after fixing the issue.
3. When IBM Storage Scale is active, auto recovery is invoked automatically to recover the disks in this node. Check the IBM Storage Scale log (/var/adm/ras/mmfs.log.latest) and the restripefs log (/var/adm/ras/restripefsOnDiskFailure.log.latest) on the file system manager node for more details.

Handling node crashes

If a failed node cannot be recovered, auto recovery migrates all data from disks in this node to other disks in cluster. If the system does not recover, delete the disks in the node and node.

1. Log in to another cluster node and run **mmlsdisk <fs-name> -M** command to get a list of disks attached to the failed node. Save the disk list in the diskList file. Each line lists a disk name.

- Run the **mmdeledisk <fsName> -F <diskList>** command to delete the disks attached to the failed node.
- Run the **mmdelnsd -F <diskList>** command to delete NSDs attached to the failed node. Run **mmdelnode** command to remove the node, or if you are replacing the node with new hardware, use the same name and IP address to continue.

To replace the failed node with a new node, start the replacement mode with the hostname and the IP address of the failed node. Install IBM Storage Scale packages and configure SSH authorization with other nodes in the cluster. Run the following command to restore IBM Storage Scale configurations in this replacement node:

```
mmsdrestore -p <cluster manager> -R <remoteFileCopyCommand> -N <replacement node>
```

Use the **mmlsmgr** command to identify the cluster manager node. Use the Remote file copy command that is configured for the cluster.

- Start IBM Storage Scale on the replacement node by running the **mmstartup -N <replacement node>** command. Confirm that IBM Storage Scale state is active by running the **mmgetstate -N <replacement node>** command.
- Prepare a stanza file to create NSDs by running the **mmcrnsd** command and add these disks into file system by running the **mmadddisk** command.

Handling multiple nodes failure

Usually, auto recovery must be enabled in an FPO cluster to protect data from multiple node failures. Set **mmchconfig restripeOnDiskFailure=yes -N all**.

However, if one file system has only two failure groups for metadata or data with default replica two, or if one file system has only 3 failure groups for metadata or data with default replica 3, auto recovery must be disabled (**mmchconfig restripeOnDiskFailure=no -N all**) in IBM Storage Scale 4.1.x, 4.2.x and 5.0.0. The issue is fixed in IBM Storage Scale 5.0.1 and later.

Usually, if the concurrent failed nodes are less than **maxFailedNodesForRecovery**, auto recovery will protect data against node failure or disk failure. If the concurrent failed nodes are larger than **maxFailedNodesForRecovery**, auto recovery exits without any action and the administrator has to take some actions to recover it.

Multiple nodes failure without SGPanic

This topic lists the steps to handle multiple nodes failure without SGPanic

- Recover the failed nodes.
- If all nodes are recovered quickly, run the **mmlsdisk <fs-name> -e** command to view the down disk list.
- Run the **mmlsnsd -X** command to check whether there are disks that are undetected by the operating system of nodes.

For example,

```
# mmlsnsd -X
Disk name      NSD volume ID      Device      Devtype   Node name      Remarks
-----
mucxs131d01   AC170E46561E7A8F   /dev/sdb    generic   mucxs131.muc.infineon.com server node
mucxs131d02   AC170E46561E7A90   /dev/sdc    generic   mucxs131.muc.infineon.com server node
mucxs531d07   AC170E4B5612838E   /dev/sdh    generic   mucxs531.muc.infineon.com server node
mucxs531d08   AC170E4B56128391   -          -         mucxs531.muc.infineon.com (not found)
server node
```

In the above output means the physical disk for the nsd mucxs531d08 is not recognized by the OS. If a disk is not detected, check the corresponding node to see if the disk is physically broken. If the undetected disks cannot be recovered quickly, remove them from the down disk list.

- Run the **mmchdisk <fs-name> start -d <down disk in step3>**.

If it succeeds, go to step5); if not, open PMR against the issue.

5. If the undetected disks cannot be recovered, run the **mmrestripefs <fs-name> -r** to fix the replica of the data whose part of replica are located in these undetected disks.

SGPanic for handling node failure

For internal disk rick storage (FPO clusters), unmountOnDiskFail must be configured as “meta”. If it is not, change the configuration by running the **mmchconfig** command.

With unmountOnDiskFail configured as meta, if you see file system SGPanic reported when nodes are non functional, there are more than three nodes with metadata disk down together or there are more than three disks with meta data down. Follow the steps in the section 8.1 to fix the issues. Run the **mmfsck -n** command to scan the file system to ensure that mmfsck displays the following message: **File system is clean finally**. If mmfsck -n does not report “File system is clean”, you need to open PMR to report the issues and fix this with guide from IBM Storage Scale.

Network switch failure

This topic describes how to handle network switch failure.

In an FPO cluster, if Auto recovery is enabled and there are more than maxFailedNodesForRecovery non functional nodes, auto recovery does not recover the nodes. By default, maxFailedNodesForRecovery is three nodes. You can change this number depending on your cluster configuration.

A switch network failure can cause nodes to be reported as non functional. If you want auto recovery to protect against switch network failures, careful planning is required in setting up the FPO cluster. For example, a network switch failure must not bring disks (with metadata) down from 3 or more failure groups, and maxFailedNodesForRecovery must be configured to a value that is larger than the number of down nodes that will result from a switch network failure.

Data locality

In an FPO cluster, if the data storage pool is enabled with **allowWriteAffinity=yes**, the data locality is decided by the following order:

- WADFG is set by **mmchattr** or the policy.
- Default WAD or WAD is set by policy and the data ingesting node.

If the file is set with WADFG, the locality complies with WADFG independent of where the data is ingested. If the file is not set with WADFG, the locality is decided according to the WAD and data-ingesting node. Also, data locality configurations are the required configurations. If there are no disks available to comply with the configured data locality, the IBM Storage Scale FPO stores the data in other disks.

The data locality might be broken if there are **mmrestripefs -r** and **mmrestripefile** after disk failure or node failure. If your applications need data locality for good performance, restore the data locality after node failure or disk failure.

Data locality impacted from down disks

All disks in a node must be configured as the same failure or locality group. After a disk is nonfunctional, **mmrestripefs -r** from auto recovery suspends the disk and restripes the data on the nonfunctional disks onto other disks in the same locality group. The data locality is not broken because the data from local disks is still in that node. If you do not have other disks available in the same locality group, **mmrestripefs -r** from auto recovery restripes the data on the nonfunctional disks onto other nodes, breaking the data locality for the applications running over that node.

Data locality impacted from the nonfunctional nodes

If a nonfunctional node does not have NSD disks in the file system, the data locality is not impacted. If the nonfunctional node has NSD disks in the file system and the node is

not recovered within dataDiskWaitTimeForRecovery (if all down disks are dataOnly disks) and metadataDiskWaitTimeForRecovery (if there is meta data NSD disk down), auto recovery suspends the disks and performs **mmrestripefs -r**. All disks from the nonfunctional nodes are not available for write and the data from the nonfunctional disks is striped onto other nodes. Therefore, the data locality is broken on the nonfunctional nodes.

Data locality impacted from unintended mmrestripefile -b or mmrestripefs -b

If the file is not set with WADFG (by policy or by mmchattr), both **mmrestripefile -b** and **mmrestripefs -b** might break the data locality.

Data Locality impacted from unintended mmrestripefile -l

If the file is not set with WADFG (by policy or by mmchattr), **mmrestripefs -l** might break the data locality. The node running **mmrestripefile -l** is considered as the data writing node and all first replica of data is stored in the data writing node for an FPO-enabled storage pool.

The following sections describe the steps to check if your data locality is broken and how fix it if needed.

Check the data locality

This topic lists the steps to check the data locality for IBM Storage Scale.

Perform the following steps to check the data locality for IBM Storage Scale releases:

- For IBM Storage Scale 4.2.2.0 and earlier, run `/usr/lpp/mmfs/samples/fpo/tsGetDataBlk`.
- For IBM Storage Scale 4.2.2.x, run `/usr/lpp/mmfs/samples/fpo/mmgetlocation`.
- For IBM Storage Scale 4.2.3, **mmgetlocation** supports the `-Y` option.

Note: `/usr/lpp/mmfs/samples/fpo/mmgetlocation` is based on `/usr/lpp/mmfs/samples/fpo/tsGetDataBlk`. Ensure that GNU GCC is installed from Linux distro before invoking `/usr/lpp/mmfs/samples/fpo/mmgetlocation`.

You can use `/usr/lpp/mmfs/samples/fpo/mmgetlocation` to query the block location of file.

You can refer the output from `/usr/lpp/mmfs/samples/fpo/mmgetlocation` about the options.

You can run `/usr/lpp/mmfs/samples/fpo/mmgetlocation -f <absolute-file-path>` to get the block location of the `<absolute-file-path>`. Also, you can run `/usr/lpp/mmfs/samples/fpo/mmgetlocation -d <absolute-dir-path>` to get the block location summary of `<absolute-dir-path>`.

For IBM Storage Scale 4.2.2.x, run `/usr/lpp/mmfs/samples/fpo/mmgetlocation`.

The following is a sample output:

```
# /usr/lpp/mmfs/samples/fpo/mmgetlocation -f /sncfs/file1G
[FILE INFO]
-----
blockSize 1024 KB
blockGroupFactor 128
metadataBlockSize 131072K
writeAffinityDepth 1
flags:
data replication: 2 max 2
storage pool name: fpodata
metadata replication: 2 max 2

Chunk 0 (offset 0) is located at disks: [ data_c8f2n04_sdg c8f2n04 ] [ data_c8f2n05_sdf
c8f2n05 ]
...
Chunk 7 (offset 939524096) is located at disks: [ data_c8f2n04_sdg c8f2n04 ] [ data_c8f2n05_sdf
c8f2n05 ]

[SUMMARY INFO]
```

```
-----
Replica num Nodename TotalChunkst
Replica 1 : c8f2n04: Total : 8
Replica 2 : c8f2n05: Total : 8
[root@c8f2n04 fpo]#
```

The summary at the end of the output shows that, for the file /sncfs/file1G, 8 chunks of the first replica are located on the node c8f2n04. The 8 chunks of the second replica are located on the c8f2n05 node.

For IBM Storage Scale 4.2.2.0 and earlier, perform the following steps to get the block location of files.

```
cd /usr/lpp/mmfs/samples/fpo/
g++ -g -DGPFSSNC_FILEMAP -o tsGetDataBlk -I/usr/lpp/mmfs/include/ tsGetDataBlk.C -L/usr/lpp/mmfs/lib/ -lgpfs
./tsGetDataBlk <filename> -s 0 -f <data-pool-block-size * blockGroupFactor> -r 3
```

Check the output of the tsGetDataBlk program:

```
[root@gpfstest2 sncfs]# /usr/lpp/mmfs/samples/fpo/tsGetDataBlk /sncfs/test -r 3
File length: 1073741824, Block Size: 2097152
Parameters: startoffset:0, skipfactor: META_BLOCK, length: 1073741824, replicas 3
numReplicasReturned: 3, numBlksReturned: 4, META_BLOCK size: 268435456
Block 0 (offset 0) is located at disks: 2 4 6
Block 1 (offset 268435456) is located at disks: 2 4 6
Block 2 (offset 536870912) is located at disks: 2 4 6
Block 3 (offset 805306368) is located at disks: 2 4 6
```

In the above example, the block size of data pool is 2 Mbytes, the blockGroupFactor of the data pool is 128. So, the META_BLOCK (or chunk) size is 2MB * 128 = 256Mbytes. Each output line represents one chunk. For example, Block 0 in the above is located in the disks with disk id 2, 4 and 6 for 3 replica.

To know the node on which the three replicas of Block 0 are located, check the mapping between disk ID and nodes:

Check the mapping between disks and nodes by **mm1sdisk** (the 9th column is the disk id of NSD) and **mm1snsd**:

disk name	driver type	sector size	failure group	holds metadata	holds data	status	avail-ability	disk id	storage pool
node1_sdb	nsd desc	512	1	Yes	No	ready	up	1	
system	desc								
node1_sdc	nsd	512	1,0,1	No	Yes	ready	up	2	
datapool									
node2_sda	nsd	512	1	Yes	No	ready	up	3	
system									
node2_sdb	nsd	512	2,0,1	No	Yes	ready	up	4	
datapool									
node6_sdb	nsd desc	512	2	Yes	No	ready	up	5	
system	desc								
node6_sdc	nsd	512	3,0,1	No	Yes	ready	up	6	
datapool									
node7_sdb	nsd	512	2	Yes	No	ready	up	7	
system									
node7_sdd	nsd	512	4,0,2	No	Yes	ready	up	8	
datapool									
node11_sdb	nsd desc	512	3	Yes	No	ready	up	9	
system	desc								
node11_sdd	nsd desc	512	1,1,1	No	Yes	ready	up	10	
datapool	desc								
node9_sdb	nsd	512	3	Yes	No	ready	up	11	
system									
node9_sdd	nsd	512	2,1,1	No	Yes	ready	up	12	
datapool									
node10_sdc	nsd desc	512	4	Yes	No	ready	up	13	
system	desc								
node10_sdf	nsd	512	3,1,1	No	Yes	ready	up	14	
datapool									
node12_sda	nsd	512	4	Yes	No	ready	up	15	
system									
node12_sdb	nsd	512	4,1,2	No	Yes	ready	up	16	

```

datapool
[root@gpfstest2 sncfs]# mmIsnsd
  File system   Disk name      NSD servers
-----
sncfs        node1_sdb    gpfstest1.cn.ibm.com
sncfs        node1_sdc    gpfstest1.cn.ibm.com
sncfs        node2_sda    gpfstest2.cn.ibm.com
sncfs        node2_sdb    gpfstest2.cn.ibm.com
sncfs        node6_sdb    gpfstest6.cn.ibm.com
sncfs        node6_sdc    gpfstest6.cn.ibm.com
sncfs        node7_sdb    gpfstest7.cn.ibm.com
sncfs        node7_sdd    gpfstest7.cn.ibm.com
sncfs        node11_sdb   gpfstest11.cn.ibm.com
sncfs        node11_sdd   gpfstest11.cn.ibm.com
sncfs        node9_sdb    gpfstest9.cn.ibm.com
sncfs        node9_sdd    gpfstest9.cn.ibm.com
sncfs        node10_sdc   gpfstest10.cn.ibm.com
sncfs        node10_sdf   gpfstest10.cn.ibm.com
sncfs        node12_sda   gpfstest12.cn.ibm.com
sncfs        node12_sdb   gpfstest12.cn.ibm.com

```

The three replicas of Block 0 are located in disk ID 2 (NSD name node1_sdc, node name is gpfstest1.cn.ibm.com), disk ID 4 (NSD name node2_sdb, node name is gpfstest2.cn.ibm.com), and disk ID 6 (NSD name node6_sdc, node name is gpfstest6.cn.ibm.com). Check each block of the file to see if the blocks are located correctly. If the blocks are not located correctly, fix the data locality.

mmgetlocation

For IBM Storage Scale 4.2.3, mmgetlocation supports the -Y option.

Synopsis

```

mmgetlocation {[-f filename] | [-d directory]}
  [-r {1|2|3|all}]
  [-b] [-L] [-1] [-Y] [--lessDetails]
  [-D [diskname,diskname,...]]
  [-N [nodename,nodename,...]]

```

Parameters

-f *filename*

Specifies the file whose block location you want to query. It should be absolute file path. For one file, the system displays the block/chunk information and the file block summary information.

-d *directory*

Specifies the directory whose block location you want to query. All files under <directory> will be checked and summarized together. <**directory**> must be the absolute directory path. The system displays one block summary for each file and one directory summary with the block information. The options -f and -d are exclusive.

Note: The sub-directories under <**directory**> won't be checked.

-r {1|2|3|all}

Specifies the replica that you want to query for the block location. 2 means replica 1 and replica 2. By default, the value is set to all.

-b

The block location is considered as file system block or as FPO chunk (file system block size * blockGroupFactor). By default, the value is set to no.

-L

Displays the detailed information (NSD ID and NSD failure group) of one block or chunk. This option impacts only the output of the block information for one file. It is not applicable when option -d is specified.

-1

Lists the NSD and total replica number on the NSD in summary of file or directory.

- Y**
Displays headers and displays the output in a colon-separated fields format.
- D {NSD[,NSD...]}**
Displays only the file block and chunk information and the summary of the specified NSDs.
- N {Node[,Node...]}**
Displays only the file block and chunk information and the summary of the specified nodes.
- lessDetails**
Does not display the file summary of each file in <directory> when option **-d** is specified. If option **-f** is specified, does not display the block details.

Notes

1. Only tested over Linux.
2. Does not recursively process the subdirectories if option **-d** is specified.
3. For FPO, if both **-D** and **-N** are specified, the **-N** option must be with only one node because no two NSDs in FPO belong to the same node.
4. For **mmgetlocation -Y**, the system displays the output in the following formats:

- a. mmgetlocation:fileSummary:filepath:blockSize:metadataBlockSize:dataReplica:metadataReplica:
storagePoolName:allowWriteAffinity:writeAffinityDepth:blockGroupFactor:(-Y -L specified)
- b. mmgetlocation:fileDataInfor:chunkIndex:offset:NSDName:NSDServer:diskID:failureGroup:
reserved:NSDName:NSDServer:diskID:failureGroup:reserved:NSDName:NSDServer:diskID:failureGr
oup:
reserved: if there are 2 or 3 replicas, repeat
"nsdName:nsdServer:diskID:failureGroup:reserved:"
if the option "-L" is not specified, the value of "diskID" and "failureGroup" will be
blank
- c. mmgetlocation:fileDataSummary:path:replicaIndex:nsdServer:nsdName:blocks:(-l specified)
mmgetlocation:fileDataSummary:path:replicaIndex:nsdServer:blocks:(-l not specified) if
there are
more than 1 NSD for replica #, each one will be output as one line if the value of
"nsdName" in
one line is "all", that means, the option "-l" is not given.
- d. mmgetlocation:dirSummary:path:replicaIndex:nsdServer:nsdName:blocks:(-l specified)

Note: If the value of **nsdName** in one line is *all*, the option **-l** is not given. So, for the option **-f**, the output is:

```
a
b
c
```

For the option **-d**, the output is:

```
c for each file
d
```

Examples

```
1 /usr/lpp/mmfs/samples/fpo/mmgetlocation -f /sncfs/file1G

[FILE /sncfs/file1G INFORMATION]
  FS_DATA_BLOCKSIZE : 1048576 (bytes)
  FS_META_DATA_BLOCKSIZE : (bytes)
  FS_FILE_DATAREPLICA : 3
  FS_FILE_METADATAREPLICA : 3
  FS_FILE_STORAGEPOOLNAME : fpodata
  FS_FILE_ALLOWWRITEAFFINITY : yes
  FS_FILE_WRITEAFFINITYDEPTH : 1
  FS_FILE_BLOCKGROUPFACTOR : 128

chunk(s)# 0 (offset 0) : [data_c3m3n03_sdd c3m3n03] [data_c3m3n02_sdc c3m3n02] [data_c3m3n04_sdc c3m3n04]
chunk(s)# 1 (offset 134217728) : [data_c3m3n03_sdd c3m3n03] [data_c3m3n04_sdc c3m3n04] [data_c3m3n02_sdc c3m3n02]
```



```

sncfs    node1_sdc  gpfstest1.cn.ibm.com
sncfs    node2_sda  gpfstest2.cn.ibm.com
sncfs    node2_sdb  gpfstest2.cn.ibm.com
sncfs    node6_sdb  gpfstest6.cn.ibm.com
sncfs    node6_sdc  gpfstest6.cn.ibm.com
sncfs    node7_sdb  gpfstest7.cn.ibm.com
sncfs    node7_sdc  gpfstest7.cn.ibm.com
sncfs    node11_sdb  gpfstest11.cn.ibm.com
sncfs    node11_sdd  gpfstest11.cn.ibm.com
sncfs    node9_sdb  gpfstest9.cn.ibm.com
sncfs    node9_sdd  gpfstest9.cn.ibm.com
sncfs    node10_sdc  gpfstest10.cn.ibm.com
sncfs    node10_sdf  gpfstest10.cn.ibm.com
sncfs    node12_sda  gpfstest12.cn.ibm.com
sncfs    node12_sdb  gpfstest12.cn.ibm.com
The three replicas of Block 0 are located in disk id 2 (NSD name node1_sdc, node name is gpfstest1.cn.ibm.com), disk id 4 (NSD name node2_sdb, node name is gpfstest2.cn.ibm.com), and disk id 6 (NSD name node6_sdc, node name is gpfstest6.cn.ibm.com). Check each block of the file to see if the blocks are located correctly. If all blocks are not located correctly, fix the data locality

```

Data locality based copy

Synopsis

```
localityCopy Device {-s {[Fileset]: Snapshot | srcDir | filePath}
                     {-t targetDir} [-l | -b] [-f] [-r]
                     [-a | -N {Node[,Node...]} | NodeFile | NodeClass]}
```

Parameters

Device

The device name of the file system to which the disks belong. File system names need not be fully-qualified. `fs0` is as acceptable as `/dev/fs0`. This must be the first parameter.

-s {[Fileset]: Snapshot | srcDir | filePath}

`Snapshot` is the snapshot name. If `:Snapshot` is specified, the global snapshot is named `Snapshot` from `Device`. If there are more than 1 snapshots existing from `:Snapshot` or `Snapshot`, it will fail. Also, if it is fileset snapshot, ensure that the fileset is linked. `srcDir` is the source directory that is copied. The directory must exist in device. If the directory is the JunctionPath of one fileset, the fileset must be linked before running the script. `filePath` is the file path that will be copied.

Note: `Snapshot` is the snapshot name. `srcDir` and `filePath` must be absolute path.

-t targetDir

Specifies the target directory to which the files from the snapshot or the directory will be copied. `targetDir` must be absolute and must exist on the node that is running the command.

-l

Only consider the locality if more than one node is involved. This might make some nodes busier than others. If there are active application jobs over the cluster and these jobs need enough network bandwidth, option `-l` makes the data copy consume as less as network bandwidth. When multiple nodes are specified with option `-N`, option `-l` might make the copy running over limited nodes and therefore take longer to finish data copy.

-b

Considers the locality if more than one node is involved and distributes the copy tasks among all involved nodes.

Note: The copy tasks are distributed at the file level (one file per copy task). The option `-l` and `-b` are exclusive. If either the option `-l` or `-b` is not specified, the option `-l` is true as default.

-f

If the to-be-copied file exists under `targetDir`, it will be overwritten if the option `-f` is specified. Or, the file will be skipped.

-r

When the option `-s {srcDir}` is specified, option `-r` will copy the files in recursive mode. For `-s {[Fileset]:Snapshot}`, option `-r` is always true.

- v**
Displays verbose information.
- a**
All nodes in the cluster are involved in copying tasks.
- N {Node[,Node...] | NodeFile | NodeClass}**
Directs a set of nodes to be involved in copying tasks. **-a** is the default if option **-N** is not specified.

Notes

1. If your file system mount point has special character, excluding +,-,_ it is not supported by this script.
2. If the file path contains special character, such as a blank character or a line break character, the file is not copied with warning.
3. When option **-a** or **-N** is specified, the file system for the **-t targetDir** must be mounted if it is from external NFS or another IBM Storage Scale file system.
4. Only copies the regular data file, does not copy link, special files.
5. If one file is not copied, the file is displayed and not copied again in the same invocation.
6. You must specify option **-s** with snapshot. For directory, the file list is not rescanned to detect any newly created files or subdirectories.

Data locality restoration

If the blocks of the file are not located as what you want, restore or change the locality of the file.

The IBM Storage Scale FPO provides interface for you to control all first replica of the blocks, all second replicas of the blocks, and all third replicas of the blocks in specific nodes. For example, you can have the first replica of all blocks located in a specific node so that the applications running over the node can read all data from local disks.

Note: The IBM Storage Scale FPO does not support the control of the location of only one or part of blocks. For example, you cannot control the location of block 1 or block 2 without changing the location of block 3.

Restoring the locality for files without WADFG

This topic lists the steps to control the first replica of all blocks.

1. Check whether the file is configured with WADFG.

```
mmlsattr -d -L /sncfs/test
file name:          /sncfs/test
metadata replication: 3 max 3
data replication:    3 max 3
immutable:          no
appendOnly:         no
flags:
storage pool name: datapool
fileset name:       root
snapshot name:
Write Affinity Depth Failure Group(FG) Map for copy:1 1,0,1
Write Affinity Depth Failure Group(FG) Map for copy:2 2,0,1
Write Affinity Depth Failure Group(FG) Map for copy:3 3,0,1
creation time:      Thu Mar 24 16:15:01 2016
Misc attributes:    ARCHIVE
Encrypted:          no
gpfs.WADFG:        0x312C302C313B322C302C313B332C302C31
```

If you see gpfs.WADFG (as per the preceding example) from the output of **mmlsattr**, the file is configured with WADFG, and, in this gpfs.WADFG case, follow the instructions in “[Restoring the locality for files with WADFG](#)” on page 729. If you do not see the gpfs.WADFG text, go to the step2.

2. Select the node to store all the blocks from the first replica of the data. One disk from this node is used to store the first replica of the file, assuming that this node has at least one local disk that serves the GPFS file system.

In IBM Storage Scale 4.2.2.0 and later, **mmrestripefile -l** is optimized to reduce unnecessary data movement. For files with WAD=1, if the target node is from the same failure group as the current node holding the replica 1 of blocks, **mmrestripefile -l** does not move the second and third replica. If it is not, **mmrestripefile -l** moves three replicas of all the blocks.

3. If you are using IBM Storage Scale 4.1.1.0 or later:

- a) ssh to the target node selected in step 2
- b) run **mmrestripefile -l filename** for each filename to set the data locality for.

If you are using IBM Storage Scale 4.1.1.0 or earlier

- a) ssh to the target node selected in step 2
- b) **mmchdisk <fs-name> suspend -d "any-one-data-disk"**
- c) **mmchdisk <fs-name> resume -d "any-one-data-disk"**
- d) **mmrestripefile -b filename**

4. Check the data locality by running:

```
/usr/lpp/mmfs/samples/fpo/tsGetDataBlk /sncfs/test -r 3
```

The first replica of all blocks is located in the target node.

Restoring the locality for files with WADFG

If you want to control the location of the first replica, second replica, and the third replica, set the WADFG attributes of the files via **mmchattr**. If you are using IBM Storage Scale 4.1.1.0 or earlier, perform these steps to restore the data locality.

1. Decide the location for data replica.
2. Run **mmchattr --write-affinity-failure-group** to set/update the new WADFG of the file Step.

In IBM Storage Scale 4.2.2.0 and later, **mmrestripefile -l** is optimized to reduce unnecessary replica data movement. For example, the original WADFG is (1;2;3). If it is changed into (4;2;3), **mmrestripefile -l** moves only the first replica of all blocks. However, if it is changed into (4), **mmrestripefile -l** might move the second and third replica. Therefore, changing the original WADFG from (1;2;3) into (4;2;3) is better than changing it into (4).

3. If you are using IBM Storage Scale 4.1.1.0 or later, skip this step. Run **mmrestripefile -l filename** or **mmrestripefile -b filename**.

In IBM Storage Scale 4.1.1.0 and later, the default option for **mmchattr** is -I yes, and the stripe function of **mmrestripefile -l** is performed when **mmchattr** is run.

4. Check the data locality.

Disk Replacement

This topic describes how to replace a disk.

Note: At any time, you cannot run the **mmrpldisk** command to replace one stopped disk used by the file system (Check the disk availability from the **mmlsdisk fs-name -L** command output. Also, the to-be-replaced disk must be up for availability).

In a production cluster, you can replace physically broken disks with new disks or replace the failed disks with new disks.

- If you have non functional disks from two failure groups for replica 3, stripe the file system to protect the data to avoid data loss from a third non functional disk from the third failure group.
- Replacing the disks is time-consuming because the whole inode space must be scanned and the IO traffic in the cluster is triggered. Therefore, schedule the disk replacement when the cluster is not busy.

The **mmrpldisk** command can be used to replace one disk in file system with a new disk and it can handle one disk in one invocation. If you want to replace only one disk, see **mmrpldisk** command.

Note: In FPO, sometimes **mmrpldisk** command does not migrate all data from the to-be-replaced disk to the newly added disk. This bug impacts IBM Storage Scale Release 3.5 and later. See the following example:

```
[root@c8f2n03 ~]# mmlsdisk sncfs -L
disk      driver   sector    failure holds   holds
name      type     size      group metadata data  status
pool      remarks
-----
n03_0      nsd      512       1 Yes     Yes   ready   up    1
system
n03_1      nsd      512       1 Yes     Yes   ready   up    2
system
n04_0      desc     512       2,0,0 Yes    Yes   ready   up    3
system
n04_1      desc     512       2,0,0 Yes    Yes   ready   up    4
system
n05_1      nsd      512       4,0,0 No     Yes   ready   up    5
system
desc
Number of quorum disks: 3
Read quorum value: 2
Write quorum value: 2

[root@c8f2n03 ~]# /usr/lpp/mmfs/samples/fpo/tsGetDataBlk /sncfs/log -s 0
File length: 1073741824, Block Size: 1048576
Parameters: startoffset:0, skipfactor: META_BLOCK, length: 1073741824, replicas 0
numReplicasReturned: 2, numBlksReturned: 8, META_BLOCK size: 134217728
Block 0 (offset 0) is located at disks: 2 5
Block 1 (offset 134217728) is located at disks: 2 3
Block 2 (offset 268435456) is located at disks: 2 5
Block 3 (offset 402653184) is located at disks: 2 3
Block 4 (offset 536870912) is located at disks: 2 5
Block 5 (offset 671088640) is located at disks: 2 3
Block 6 (offset 805306368) is located at disks: 2 5
Block 7 (offset 939524096) is located at disks: 2 3

[root@c8f2n03 ~]# mmrpldisk sncfs n03_1 n03_4
[root@c8f2n03 ~]# mmlsdisk sncfs -L
disk      driver   sector    failure holds   holds
name      type     size      group metadata data  status
pool      remarks
-----
n03_0      nsd      512       1 Yes     Yes   ready   up    1
system
n04_0      desc     512       2,0,0 Yes    Yes   ready   up    3
system
n04_1      nsd      512       2,0,0 Yes    Yes   ready   up    4
system
n05_1      nsd      512       4,0,0 No     Yes   ready   up    5
system
n03_4      desc     512       1 Yes     Yes   ready   up    6
system
nsd
Number of quorum disks: 3
Read quorum value: 2
Write quorum value: 2

[root@c8f2n03 ~]# /usr/lpp/mmfs/samples/fpo/tsGetDataBlk /sncfs/log -s 0
File length: 1073741824, Block Size: 1048576
Parameters: startoffset:0, skipfactor: META_BLOCK, length: 1073741824, replicas 0
numReplicasReturned: 2, numBlksReturned: 8, META_BLOCK size: 134217728
Block 0 (offset 0) is located at disks: 6 5
Block 1 (offset 134217728) is located at disks: 1 3
Block 2 (offset 268435456) is located at disks: 1 5
Block 3 (offset 402653184) is located at disks: 1 3
Block 4 (offset 536870912) is located at disks: 6 5
Block 5 (offset 671088640) is located at disks: 6 3
Block 6 (offset 805306368) is located at disks: 1 5
Block 7 (offset 939524096) is located at disks: 6 3
After replacing n03_1 with n03_4, part of data located in n03_1 are migrated into n03_4 and others are migrated into n03_0. Therefore, mmrpldisk doesn't mean copy data from the to-be-replaced disks into new added disks. mmrpldisk might break the data locality and you need to see the Section 9 to restore data locality if needed.
```

If you want to replace more than one disk, run the **mmrpldisk** command multiple times. The PIT job is triggered to scan the whole inode space to migrate the data to disks that are going to be replaced. The IO traffic is triggered and is time-consuming if you have to run the **mmrpldisk** command multiple times. To

speed the replacement process, see the following sub sections to replace more than one disk in the file system.

Replace more than one active disks

This topic describes how to replace more than one active disk.

If you want to replace more than one disk used in file system, and if you have a lot of files or data in the file system, it will take long time to do this if you are using the **mmrpldisk** command for each disk.

If you have additional idle disk slots, you can plug new disks into these idle slots and run **mmcrlsd** to create new NSD disks against the disks that are to be added, run **mmadddisk** (without the option -r) to add the new disks into the file system and then **mmdeldisk** the disks that are to be replaced by using **mmdeldisk**.

Note: If you place new disks in the same failure group of the disks that are to be replaced, the above operations will maintain the data locality for the data from disks that are to be replaced. IBM Storage Scale keeps the data in the original failure group.

If you do not have additional idle disk slots, run the **mmdeldisk** command on the disks that are to be replaced, run **mmcrlsd** to create the NSD disks and run **mmadddisk** to add the NSD disks to the file system. You might have to run **mmrestripefs -b** to balance the file system but this breaks the data locality.

Replace more than one broken disks

If you want to replace more than one disk that are physically broken, you cannot read any data from these disks or write any data into these disks,

if the broken disks have been restriped they become emptied or non functional. Run the **mmdeldisk** command directly. Pull out the broken disks, pull in the new disks, run **mmcrlsd**, and then run **mmadddisk** to add them into the file system.

If the broken disks have not been restriped (then, it might be ready/down or ready/up), then take the following steps:

1. Disable auto recovery temporarily (refer the section 2.1, step 2)
2. Pull out the broken disks directly. You could run **mmllnsd -X** to check what these pulled-out disks will be like: node7_sdn C0A80A0756FBAA89 - - gpfstest7.cn.ibm.com (not found) server node
3. Pull in the new disks.
4. **mmcrlsd** for the new disks (take new NSD name)
5. **mmadddisk <fs-name> -F <new-nsd-file from step4>**
6. To delete the broken disks from the file system, see *Disk media failure in IBM Storage Scale: Problem Determination Guide*.

Auto recovery

The FPO-enabled/disabled storage pool over internal disks are subject to frequent node and disk failures because of the commodity hardware used in IBM Storage Scale clusters.

IBM Storage Scale auto recovery feature is designed to handle random but routine node and disk failures without requiring manual intervention. However, auto recovery cannot cover all catastrophic outages involving large number of nodes and disks at once. Administrator assessment of the situation and judgment is required to determine the cluster recovery action.

Note:

Following are some important recommendations:

- In IBM Storage Scale 5.0.2 and later, the suspended disks are resumed automatically when the node rejoins the cluster. In versions earlier than 5.0.2, the system administrator must issue a command like the following one to resume the disks:

```
mmchdisk <FileSystem> resume -a
```

- If extended outages (days and weeks) are expected, it is recommended to remove that node and all associated disks from the cluster to avoid this outage from affecting subsequent recovery actions.
- If the failed disk is meta disk, during auto recovery, GPFS will try to suspend the failed disk using the **mmchdisk <file-system>** command. If the remaining failure groups of meta or data disks is less than the value of -r/-m, this will make **mmchdisk <file-system>** suspend/fail, and therefore auto recovery will not take further actions.

Failure and recovery

There are two main failures to consider for FPO environments.

1. Node failure and outages: These outages include reboot, kernel crash and hang and can last long. When a node is inaccessible, all the associated disks also become inaccessible.
2. Disk failures: These failures include disk failures, hard IO errors and are generally triggered by hardware failures and affect specific disks.

IBM Storage Scale recovery actions are enabled by setting the `restripeOnDiskFailure` configuration option to yes. When this option is enabled, auto recovery leverages the IBM Storage Scale event callback mechanism to trigger necessary actions to perform recovery actions. Specifically, the following system callbacks are installed when `restripeOnDiskFailure=yes`:

- **event = diskFailure action: /usr/lpp/mmfs/bin/mmcommon recoverFailedDisk %fsName %diskName**
- **event = nodeJoin action: /usr/lpp/mmfs/bin/mmcommon restartDownDisks %myNode %clusterManager %eventNode**
- **event = nodeLeave action: /usr/lpp/mmfs/bin/mmcommon stopFailedDisk %myNode %clusterManager %eventNode**

Important: Disable auto recovery while you are doing any planned maintenance such as upgrading an operating system, upgrading hardware or firmware, or doing any of the following tasks for IBM Storage Scale: installing a later version, deleting a node, or deleting or replacing an NSD server. Otherwise auto recovery, which handles unexpected disk or node exceptions, can interfere with or break the maintenance process.

diskFailure Event

This event is triggered when a disk I/O operation fails. Upon I/O failure, IBM Storage Scale marks the disk from read/up to ready/down. This I/O failure can also be caused by a node, because all disks connected by the node become unavailable, or a disk failure.

The disk state is ready/down.

Recovery process

1. Perform simple checks, such as fpo pool and replication >1.
2. Check the **maxDownDisksForRecovery** (default 16), **maxFailedNodesForRecovery** (default 3). Abort if the limit is exceeded. Note that these limits can be changed by using the configuration parameters.
3. If the number of failed FGs is less than 2, wait until **dataDiskWaitTimeForRecovery** (default 3600/2400) expires, otherwise wait for **minDiskWaitTimeForRecovery** (default 1800 sec) to expedite recovery due to increased risk.
4. If the available FGs for metadata is less than three, no action is taken because recovery cannot be performed due to the metadata outage.

5. After the recovery wait period has passed, recheck the node and disk availability status to ensure that recovery actions are taken.
6. Suspend all the failed and unavailable disks by running the **tschdisk suspend** command.
7. Restripe the data. If a previous restripe process is running, stop it and start a new process.
8. At successful completion, disks will be in suspended/down or suspended/up if the node is recovered during the restripe.

Note: If the file system version is 5.0.2 and later, the suspended disks from auto recovery are resumed when the node with suspended or to be emptied disks joins the cluster again. If the file system version is earlier than 5.0.2, cluster administrator has to manually run `mmchdisk fs-name resume -a` to resume the disks.

nodeJoin Event

This event is triggered when a node joins the cluster after a node reboot or rejoined after losing membership to the cluster or getting started after an extended outage. Scope of the recovery is all file systems to which the node disks might belong to. In most case, the disk state can be ready/up if no I/O operation has been performed or ready/down. However, based on the prior events, the state could vary to suspended/down or unrecovered/recovering.

Recovery process

1. Perform simple checks on the disks assigned to the file systems.
2. Check if a **tschdisk start** is already running from a prior event. Kill the process to include disks from the current nodes.
3. Start all disks on all nodes by running: **tschdisk start -a** to optimize recovery time. This command requires all nodes in the cluster to be functioning in order to access all the disks in the file system.
4. Start All down disks on all Active nodes by running: **tschdisk start -F<file containing disk list>**.
5. If the file system version is 5.0.2 and later, auto recovery will run `mmchdisk fs-name resume -d <suspended-disk-by-auto-recovery>`. If the file system version is earlier than 5.0.2, this command will not be executed.
6. After successful completion, for file system version 5.0.2 and later, all disks must be in the ready/up state. For file system version earlier than 5.0.2, all disks must be in the suspended/up state.

For file system version 5.0.2 and later, if the administrator runs `mmchdisk fs-name suspend -d <disks>` and these disks do not resume by auto recovery, the administrator needs to resume these disks manually.

If a new diskFailure event is triggered while **tschdisk start** is in progress, the disks will not be restored to the Up state until the node joins the cluster and triggers a nodeJoin event.

nodeLeave Event

This event is triggered when a node leaves the cluster, is expelled, or shut down.

The processing of this event is similar to the diskFailure event, except that disks may not already be marked as Down when this event is received. Note that a diskFailure event can still be generated based on an I/O activity in the cluster. If it is generated, no action will be taken by the diskFailure event handler if the owning node is also down, thereby allowing the nodeLeave event to control the recovery. In most cases, the disk state could be ready/up if no I/O operation has been performed or ready/down. However, based on prior events the state could be suspended/down or unrecovered/recovering.

Recovery process

1. Wait for the specified duration to give the failed nodes a chance to recover.

2. Check the Down nodes count, Down disks count and available data and metadata FG count to check against the maximum limit.
3. Build a list of disk to act upon, ignoring suspended, empty, to be emptied.
4. Run **tsrestripefs** to restore replica count to the stated values.
5. After successful completion, disks can be in the suspended/down state or no action may be taken if the nodeJoin event is triggered within the **recoveryWaitPeriod**.

QoS support for autorecovery

When QoS is configured, the autorecovery process runs in a QoS class.

To get QoS support for autorecovery, you must enable QoS and assign IOPS to the maintenance and other classes of the storage pools that you want autorecovery to restore. For more information, see [“Setting the Quality of Service for I/O operations” on page 237](#).

In IBM Storage Scale 4.2.1.x and 4.2.2.x, the autorecovery process always runs in the QoS maintenance class. If you assign a smaller share of IOPS to the maintenance class, this setting ensures that autorecovery does not compete with normal processes for I/O operations.

In IBM Storage Scale 4.2.3 and later, the autorecovery process runs in the QoS maintenance class only if one replica is lost. If more than one replica is lost, the autorecovery process runs in the QoS other class so that it completes faster.

Restrictions

An FPO environment includes restrictions.

The following restrictions apply:

- Storage pool properties can be set only when the pool is created and cannot be changed later.
- All disks in an FPO pool must be assigned an explicit failure group.
- All disks in an FPO pool must have exactly one NSD server associated with them.
- All disks in an FPO pool that share an NSD server must belong to the same failure group.
- When replacing a disk in an FPO pool, the old and new disks must have the same NSD server.
- Disks must be removed from the file system before NSD servers can be changed.
- FPO is not supported on Windows and ZLinux.
- FPO is not supported on ECE.

There might be additional limitations and restrictions. For the latest support information, see the [IBM Storage Scale FAQ in IBM Documentation](#).

Chapter 49. Encryption

GPFS provides support for file encryption that ensures both secure storage and secure deletion of data. GPFS manages encryption through the use of encryption keys and encryption policies.

Note: File encryption is available with IBM Storage Scale Advanced Edition, IBM Storage Scale Data Management Edition, or IBM Storage Scale Developer Edition or IBM Storage Scale Erasure Code Edition. The file system must be at GPFS 4.1 or later. Encryption is supported in the following environments:

- Multicluster environments (provided that the remote nodes have their own /var/mmfs/etc/RKM.conf files and access to the remote key management servers. For more information, see [“Encryption keys” on page 735](#).)
- FPO environments

Secure storage uses encryption to make data unreadable to anyone who does not possess the necessary encryption keys. The data is encrypted while “at rest” (on disk) and is decrypted on the way to the reader. Only data, not metadata, is encrypted.

GPFS encryption can protect against attacks targeting the disks (for example, theft or acquisition of improperly discarded disks) as well as attacks performed by unprivileged users of a GPFS node in a multi-tenant cluster (that is, a cluster that stores data belonging to multiple administrative entities called tenants). However, it cannot protect against deliberate malicious acts by a cluster administrator.

Secure data deletion leverages encryption and key management to guarantee erasure of files beyond the physical and logical limitations of normal deletion operations. If data is encrypted, and the master key (or keys) required to decrypt it have been deleted from the key server, that data is effectively no longer retrievable. See [“Encryption keys” on page 735](#).

Important: Encryption should not be viewed as a substitute for using file permissions to control user access.

Encryption keys

GPFS uses the following types of encryption keys:

master encryption key (MEK)

An MEK is used to encrypt file encryption keys.

MEKs are stored in remote key management (RKM) servers and are cached by GPFS components. To ensure the currency of the cached keys and that they are not removed from the server, the key cache is periodically refreshed from the RKM servers according to the value of the **encryptionKeyCacheExpiration** parameter. For more information, see *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide*. GPFS receives information about the RKM servers in a separate /var/mmfs/etc/RKM.conf configuration file. Encryption rules present in the encryption policy define which MEKs should be used, and the /var/mmfs/etc/RKM.conf file provides a means of accessing those keys. The /var/mmfs/etc/RKM.conf also specifies how to access RKMs containing MEKs used to encrypt files created under previous encryption policies.

An MEK is identified with a unique *Keyname* that combines the name of the key and the RKM server on which it resides. See [“Encryption policy rules” on page 736](#) for *Keyname* format.

file encryption key (FEK)

An FEK is used to encrypt sectors of an individual file. It is a unique key that is randomly generated when the file is created. For protection, it is encrypted (or “wrapped”) with one or more MEKs and stored in the *gpfs*.Encryption extended attribute of the file.

A wrapped FEK cannot be decoded without access to the MEK (or MEKs) used to wrap it. Therefore, a wrapped FEK is useless to an attacker and does not require any special handling at object deletion

time. If necessary, an FEK can be rewrapped using a new set of MEKs to allow for operations like MEK expiration and rotation, compromised key removal, and data expiration.

Note: If an encryption policy specifies that an FEK be wrapped multiple times, only one of the wrapped-FEK instances needs to be unwrapped for the file to be accessible.

Encryption policies

An encryption policy consists of a set of policy rules for one of two purposes: managing the encryption of a group of files or re-wrapping the file encryption keys of already encrypted files.

The following encryption policy rules are available:

- The **ENCRYPTION IS** rule specifies how a file is to be encrypted and how file encryption keys (FEKs) are to be wrapped (that is, encrypted) with master encryption keys (MEKs).
- The **SET ENCRYPTION** rule describes a group of files to be encrypted and specifies the type of encryption (as defined by an earlier **ENCRYPTION IS** rule) that is to be done.
- The **SET ENCRYPTION EXCLUDE** command signals the end of a series of **SET ENCRYPTION** rules.
- The **CHANGE ENCRYPTION KEYS** rule re-wraps FEKs. FEKs that were previously wrapped with a specified MEK are unwrapped and then re-wrapped with a new MEK.

The first three types of rules appear in policies to encrypt files. The fourth type of rule typically appears in a policy by itself or with other **CHANGE ENCRYPTION KEYS** rules.

Encryption policies are configured with the **mmchpolicy** command. A policy for re-wrapping FEKs is applied with the **mapplypolicy** command. A policy for encrypting a set of files is applied whenever a file is created or is restored from backup.

When a file is created or is restored, the following steps occur:

1. IBM Storage Scale evaluates the rules in the policy sequentially. The type of processing depends on the type of rule:
 - For an **ENCRYPTION IS** rule, the encryption specification is saved for future use.
 - For a **SET ENCRYPTION** rule, if the created or restored file does not match the file description in the rule, the rule is skipped and processing goes on to the next rule. If the file does match the file description in the rule, encryption is postponed until the entire policy is scanned.
 - If a **SET ENCRYPTION EXCLUDE** command is encountered, evaluation of the rules stops.
Note: Evaluation of the rules also stops if the end of the of the policy is reached or if the file matches the file description of eight **SET ENCRYPTION** rules. Eight is the maximum number of **SET ENCRYPTION** rules that can be applied to one file.
2. After the encryption policy is evaluated, a FEK is generated and the file is encrypted with it.
3. Then the FEK is wrapped separately for each of the **SET ENCRYPTION** rules that the file matched. For example, if the file matched three **SET ENCRYPTION** rules, then three separate wrappings of the FEK are created. The wrapped FEKs are stored in the **gpfs.Encryption** extended attribute of the file. Only one of the wrapped FEKs needs to be unwrapped to access the file.

Notes:

1. When an encryption policy is changed, the changes apply only to the encryption of subsequently created files.
2. Encryption policies are defined on a per-file-system basis by a system administrator. After the encryption policies are put in place, they can result in files in different filesets or with different names being encrypted differently.

Encryption policy rules

In many respects encryption policy rules are handled like file placement rules:

- An encryption policy that defines the encryption of new files and restored files must be installed into IBM Storage Scale with the **mmchpolicy** command.
- An encryption policy is applied with the **mmaplypolicy** command.
- When a file is created or restored, the encryption policy determines whether the file is to be encrypted and how it is to be encrypted.
- Existing files are not encrypted. To encrypt a file that is currently not encrypted, copy it into a new file whose encryption policy rules dictate that the file is to be encrypted. Note that renaming a file does not change its encryption policy. The encryption policy is defined at the time that the file is created.

For more information, see “[Encryption policies](#)” on page 736 and “[Overview of policies](#)” on page 531.

GPFS provides the following rules with which you can specify encryption policies:

ENCRYPTION IS

This rule is used to specify how a file is to be encrypted and how the FEK is to be wrapped.

The syntax of the ENCRYPTION IS rule is:

```
RULE 'RuleName' ENCRYPTION 'EncryptionSpecificationName' IS
      ALGO 'EncParamString'
      COMBINE 'CombineParamString'
      WRAP 'WrapParamString'
      KEYS('Keyname'[, 'Keyname', ... ])
```

where:

ALGO EncParamString

specifies the encryption parameter string, which defines the following:

- encryption algorithm
- key length
- mode of operation
- key derivation function

The following encryption parameter strings are valid:

Table 52. Valid EncParamString values

Value	Description
AES:128:XTS:FEK:HMACSHA512	Encrypt the file with AES in XTS mode. The FEK is 128 bits long and is preprocessed using HMAC with SHA-512.
AES:256:XTS:FEK:HMACSHA512	Encrypt the file with AES in XTS mode. The FEK is 256 bits long and is preprocessed using HMAC with SHA-512.
AES:128:CBC:FEK:HMACSHA512	Encrypt the file with AES in CBC mode. The FEK is 128 bits long and is preprocessed using HMAC with SHA-512.
AES:192:CBC:FEK:HMACSHA512	Encrypt the file with AES in CBC mode. The FEK is 192 bits long and is preprocessed using HMAC with SHA-512.
AES:256:CBC:FEK:HMACSHA512	Encrypt the file with AES in CBC mode. The FEK is 256 bits long and is preprocessed using HMAC with SHA-512.

COMBINE CombineParamString

specifies a string that defines the mode to be used to combine MEKs specified by the KEY statement.

The following combine parameter string values are valid:

Table 53. Valid combine parameter string values	
Value	Description
XORHMACSHA512	Combine MEKs with a round of XOR followed by a round of HMAC with SHA-512.
XOR	Combine MEKs with a round of XOR.

WRAP WrapParamString

specifies a string that defines the encryption algorithm and the wrapping mode to be used to wrap the FEK.

The following wrapping parameter string values are valid:

Table 54. Valid wrapping parameter string values	
Value	Description
AES:KWRAP	Use AES key wrap to wrap the FEK.
AES:CBCIV	Use AES in CBC-IV mode to wrap the FEK.

KEYS ('Keyname'[, 'Keyname', ...])

specifies one or more keys to be applied. Each *Keyname* is a unique identifier that combines the name of the key and the RKM server on which it resides. The format for *Keyname* is:

KeyId:*RkmId*

where

KeyId

An internal identifier that uniquely identifies the key inside the RKM. Valid characters for *KeyId* are the following: 'A' through 'Z'; 'a' through 'z'; '0' through '9'; and '-' (hyphen). The minimum length of *KeyId* is one character; the maximum length is 60 characters.

RkmId

The identifier of the /var/mmfs/etc/RKM.conf entry for the RKM that manages the key. An RKM ID must be unique within the cluster, must be 1-21 characters in length, and can contain only the characters a - z, A - Z, 0 - 9, or underscore (_). The first character cannot be a numeral.

Notes:

1. The maximum number of keys you can specify with the ENCRYPTION IS rule is eight.
2. The number of keys that can be used to encrypt a single file is permanently limited by the inode size of the file system.
3. You cannot specify the same key more than once in a given ENCRYPTION IS rule. Also, do not specify keys with identical values in an ENCRYPTION IS rule. Specifying the same key or identically-valued keys could result in a security breach for your data.

SET ENCRYPTION

The SET ENCRYPTION rule is similar to the SET POOL rule. If more than one such rule is present, all SET ENCRYPTION rules are considered and the FEK is wrapped once for each of the rules that apply (up to the maximum of eight). As mentioned in “Encryption keys” on page 735, if an FEK is wrapped multiple times, only one of the wrapped-FEK instances needs to be unwrapped for the file to be accessed.

If no SET ENCRYPTION rule is applicable when a file is created, the file is not encrypted.

The syntax of the SET ENCRYPTION rule is:

```
RULE 'RuleName' SET ENCRYPTION 'EncryptionSpecificationName' [,  
  'EncryptionSpecificationName', ...]  
  [FOR FILESET ('FilesetName', 'FilesetName')...]  
  [WHERE SqlExpression]
```

where:

EncryptionSpecificationName

is the name of a specification defined by an ENCRYPTION IS rule.

To stop traversing policy rules at a certain point and encrypt using only those rules that have matched up to that point, use the SET ENCRYPTION EXCLUDE rule:

```
RULE ['RuleName'] SET ENCRYPTION EXCLUDE  
  [FOR FILESET ('FilesetName', 'FilesetName')...]  
  [WHERE SqlExpression]
```

Note: Encryption policies do not support the ACTION clause.

Default encryption parameters

In addition to the values that are shown in [Table 52 on page 737](#), the ALGO parameter can also be followed by one of the following default values:

- DEFAULTNISTSP800131A

This value is equivalent to the following parameters:

```
ALGO 'AES:256:XTS:FEK:HMACSHA512'  
COMBINE 'XORHMACSHA512'  
WRAP 'AES:KWRAP'
```

- DEFAULTNISTSP800131AFAST

This value is equivalent to the following parameters:

```
ALGO 'AES:128:XTS:FEK:HMACSHA512'  
COMBINE 'XORHMACSHA512'  
WRAP 'AES:KWRAP'
```

The two default values have almost equivalent effects. The only difference is in the resulting length of the FEK. The FEK is 256 bits in the first default value but 128 bits in the second one. Of the two default values, DEFAULTNISTSP800131A is the better choice in most situations, because the 256-bit FEK provides better security. However, because of its shorter FEK, DEFAULTNISTSP800131AFAST provides a 5 - 20% speedup in workloads that involve large block random reads and direct I/O. It is available in the following IBM Storage Scale releases:

- 5.0.1 and later
- 5.0.0 with APAR IJ04786
- 4.2.3 with APAR IJ04788
- 4.1.1 with APAR IJ04789

Before you apply an encryption rule that contains -- DEFAULTNISTSP800131AFAST, ensure that all the nodes are at the required release or APAR number.

The following example shows the use of -- DEFAULTNISTSP800131A:

```
RULE 'somerule' ENCRYPTION 'somename' IS  
ALGO 'DEFAULTNISTSP800131A'  
KEYS('KEY-2f1f7700-de74-4e55-a9be-bee49c5b3af8:RKMKMIP3')
```

Do not use the COMBINE parameter or the WRAP parameter in the same rule with -- DEFAULTNISTSP800131A or -- DEFAULTNISTSP800131AFAST.

Example of an encryption policy

This is an example of an encryption policy:

```
RULE 'myEncRule1' ENCRYPTION 'E1' IS
    ALGO 'DEFAULTNISTSP800131A'
    KEYS('1:RKM_1', '2:RKM_2')

RULE 'myEncRule2' ENCRYPTION 'E2' IS
    ALGO 'AES:256:XTS:FEK: HMACSHA512'
    COMBINE 'XOR'
    WRAP 'AES:KWRAP'
    KEYS('3:RKM_1')

RULE 'myEncRule3' ENCRYPTION 'E3' IS
    ALGO 'AES:128:CBC:FEK:HMACSHA512'
    COMBINE 'XORHMACSHA512'
    WRAP 'AES:CBCIV'
    KEYS('4:RKM_2')

RULE 'Do not encrypt files with extension enc4'
    SET ENCRYPTION EXCLUDE
    FOR FILESET('fs1')
    WHERE NAME LIKE '%.enc4'

RULE 'Encrypt files with extension enc1 with rule E1'
    SET ENCRYPTION 'E1'
    FOR FILESET('fs1')
    WHERE NAME LIKE '%.enc1'

RULE 'Encrypt files with extension enc2 with rule E2'
    SET ENCRYPTION 'E2'
    FOR FILESET('fs1')
    WHERE NAME LIKE '%.enc2'

RULE 'Encrypt files with extension enc* with rule E3'
    SET ENCRYPTION 'E3'
    FOR FILESET('fs1')
    WHERE NAME LIKE '%.enc%'
```

Note:

In this example encryption policy:

- All files in fileset `fs1` are treated as follows:
 - If the extension is equal to `enc4`, the file is not encrypted. This happens because the `ENCRYPTION EXCLUDE` rule is matched first, stopping the traversal of the remaining rules before any additional matches can be made.
 - If the extension is equal to `enc1`, the file is encrypted with a 256-bit FEK, using AES in XTS mode; the FEK is preprocessed with HMAC with SHA-512, and the FEK is then wrapped twice:
 - once with AES key wrap, with keys `1:RKM_1` and `2:RKM_2` combined via one round of XOR followed by one round of HMAC with SHA-512
 - once with AES in CBC-IV mode using key `4:RKM_2`This happens because both rules `E1` and `E3` apply, since extension `enc1` matches both `%.enc1` and `%.enc%`. Note that the encryption algorithms specified by rule `E1`, which grant a stronger security than those of rule `E3`, are chosen and applied.
 - If the extension is equal to `enc2`, the file is encrypted with a 256-bit FEK, using AES in XTS mode; the FEK is preprocessed with HMAC with SHA-512; and the FEK is then wrapped twice:
 - once with AES key wrap using key `3:RKM_1`
 - once with AES in CBC-IV mode using key `4:RKM_2`This happens because both rules `E2` and `E3` apply, since extension `enc2` matches both `%.enc2` and `%.enc%`.
 - If the extension is equal to `enc3`, the file is encrypted with a 128-bit FEK, using AES in CBC mode; the FEK is preprocessed with HMAC with SHA-512; and the FEK is then wrapped once with AES in CBC-IV mode using key `4:RKM_2`.

This happens because only rule E3 applies, since extension enc3 only matches %.enc%.

- A GPFS node with access to both keys 1:RKM_1 and 2:RKM_2 or to key 4:RKM_2 can access a file with extension enc1.
- A GPFS node with access to key 3:RKM_1 or to key 4:RKM_2 can access a file with extension enc2.
- A GPFS node with access to key 4:RKM_2 can access a file with extension enc3.
- No key is required to access a file with extension enc4.
- A file with extension enc1 is securely deleted when either key 1:RKM_1 or 2:RKM_2 and key 4:RKM_2 are destroyed in their respective RKMs (and their cached copies have been flushed).
- A file with extension enc2 is securely deleted when key 3:RKM_1 and key 4:RKM_2 are destroyed in their respective RKMs (and their cached copies have been flushed).
- A file with extension enc3 is securely deleted when key 4:RKM_2 is destroyed in its respective RKM (and its cached copies have been flushed).
- Once created, a file may not be encrypted with more MEKs, only with different MEKs using the REWRAP rule.

Rewrapping policies

Rewrapping policies are policies that change how a set of FEKs is encrypted by changing the set of MEKs that wrap the FEKs. Rewrapping applies only to files that are already encrypted, and the rewapping operation acts only on the `gpfs.Encryption EA` of the files. Rewrapping is done by using the `mmaplypolicy` command to apply a set of policy rules containing one or more `CHANGE ENCRYPTION KEYS` rules. These rules have the form:

```
RULE 'ruleName' CHANGE ENCRYPTION KEYS FROM 'Keyname_1' to 'Keyname_2'  
[FROM POOL 'poolName']  
[FOR FILESET(...)]  
[SHOW(...)]  
[WHERE ... ]
```

where:

- `Keyname_1` is the unique identifier of the MEK to be replaced. (See “[Encryption policy rules](#)” on page [736](#) for `Keyname` format.)
- `Keyname_2` is the unique identifier of the new MEK, which replaces the old MEK identified by `Keyname_1`.
- The `FOR FILESET` and `WHERE` clauses narrow down the set of affected files.

Both `Keyname_1` and `Keyname_2` are listed, and only the files that currently use `Keyname_1` have their FEKs rewrapped with `Keyname_2`. Files that do not currently use `Keyname_1` are not affected by the operation.

Notes:

1. Only the *first* matching `CHANGE ENCRYPTION KEYS` rule is applied to each file. The rule rewraps each wrapped version of the FEK that was encrypted with the MEK in the `CHANGE ENCRYPTION KEYS` rule.
2. The same MEK cannot be used more than once in a particular wrapping of the FEK.

Tip: The `mmaplypolicy` command always begins by scanning all of the files in the affected file system or fileset to discover files that meet the criteria of the policy rule. In the preceding example, the criterion is whether the file is encrypted with an FEK that is wrapped with the MEK `Keyname_1`. If your file system or fileset is very large, you might want to delay running `mmaplypolicy` until a time when the system is not running a heavy load of applications. For more information, see the topic “[Phase one: Selecting candidate files](#)” on page [553](#).

Preparation for encryption

Preparing for encryption includes verifying the version of IBM Storage Scale, installing a remote encryption key server, preparing the cluster, and preparing the encryption key server back ends.

[“Terms defined” on page 742](#)

[“Required software: IBM Storage Scale” on page 743](#)

[“Required software: Remote Key Management \(RKM\) server” on page 743](#)

[“Preparing your cluster for encryption” on page 743](#)

[“Preparing the remote key management \(RKM\) server” on page 744](#)

[“RKM back ends” on page 744](#)

[“The RKM.conf file and the RKM stanza” on page 744](#)

[“Adding backup RKM servers in a high-availability configuration” on page 746](#)

[“The client keystore directory and its files” on page 746](#)

Terms defined

The following terms are important:

device group

See *tenant*.

file encryption key

A *file encryption key (FEK)* is a key for encrypting file data. See [“Encryption keys” on page 735](#).

host

See *tenant*.

key client

A *key client* is an entity in the cluster that represents the nodes that access encrypted files. The key client receives master encryption keys (MEKs) from the tenant of the Remote Key Management (RKM) server.

key server or Remote Key Management (RKM) server

A *key server* is a server that authenticates key clients and provides them with MEKs. Examples of key server software products are IBM Security Key Lifecycle Manager (SKLM) and Thales Vormetric Data Security Manager (DSM).

master encryption key

A *master encryption key (MEK)* is a key for encrypting FEKs.

tenant

A *tenant* is an entity on a key server that contains MEKs and certificates.

- In SKLM, a tenant is called a *device group*.
- In DSM, a tenant is called a *host*.
- The **mmkeyserv** command uses the generic keyword **tenant**.

Simplified setup and regular setup

The *simplified setup* is a method for configuring the encryption environment in which you use the **mmkeyserv** command to configure and manage cluster-wide encryption configuration. This method is much preferred if your key server is SKLM or HashiCorp Vault Enterprise because the **mmkeyserv** command automatically performs many of the steps that must be done manually in the regular setup. You can use this setup method only with SKLM or HashiCorp Vault Enterprise KMIP Secrets Engine.

Note: HashiCorp Vault KMIP Secrets Engine is available only with the HashiCorp Vault Enterprise package.

The *regular setup* is a method for configuring the encryption environment in which you generate client credentials and then manually edit and distribute encryption configuration files to the nodes in the cluster. You can use this setup method with either SKLM or DSM.

Note:

- See the remaining subtopics in this help topic for number of important differences between the two methods.
- For more information, see the instructions for the simplified setup and regular setup in the help topics that follow this topic.

Required software: IBM Storage Scale

The following table lists the versions of IBM Storage Scale that support encryption and the encryption setup methods:

Table 55. Required version of IBM Storage Scale		
IBM software	Version	Encryption setup
IBM Storage Scale <ul style="list-style-type: none">• Advanced Edition• Data Management Edition• Erasure Code Edition	4.1 or later	<ul style="list-style-type: none">• Regular setup• Regular setup with certificate chain
	4.2.1 or later	Simplified setup
	5.1.6 or later	Simplified setup for HashiCorp Vault KMIP Secrets Engine

Required software: Remote Key Management (RKM) server

The next table shows the RKM server software that IBM Storage Scale supports.

Table 56. Remote Key Management servers		
RKM server	Version	Type of encryption setup
IBM Security Key Lifecycle Manager	2.6 or later	<ul style="list-style-type: none">• Simplified setup• Simplified setup with certificate chain
Thales Vormetric Data Security Manager	6.2 or later	Regular setup
HashiCorp Vault Enterprise	1.12 or later	Simplified setup
Thales CipherTrust Manager	2.5.x and 2.8 or later	Regular setup

¹ For more information, see the Question 2.15 in [IBM Storage Scale FAQ Documentation](#), "What are the requirements/limitations for using native encryption in IBM Storage Scale Advanced Edition, Data Management Edition, or Erasure Code Edition?"

Preparing your cluster for encryption

Follow these steps:

1. Verify the following items in your IBM Storage Scale cluster:
 - The cluster is running the correct version of IBM Storage Scale and the correct version of a supported RKM server. These versions are listed in [Table 55 on page 743](#) and [Table 56 on page 743](#).
 - The file system daemon is running.
2. Ensure that the following packages are installed:
 - `gpfs.gskit`

- `gpfs.crypto`
3. Set up an IBM Storage Scale file system on the cluster. The version of the file system must be IBM Storage Scale Release 4.1 or later. Configure the following features on the file system:

- a. Create the file system with the inode size of 4 KiB. Choosing this minimum size is a good idea because 4 KiB accommodates the GPFS encryption extended attribute that is assigned to each encrypted file at file creation time. This extended attribute contains one or more wrapped FEKs so it can potentially grow large. For more information, see “[Encryption policies](#)” on page 736.
- b. Enable fast extended attributes. This setting is the default for a newly created file system if you are running 4.1 or later. To verify that fast extended attributes are enabled, issue the following command, where `<Device>` is the name of the file system:

```
mmlsfs <Device> --fastea
```

However, if your file system was migrated from an earlier level, issue the following command to add support for fast extended attributes:

```
mmigrate <Device> --fastea
```

For more information, see *Completing the migration to a new level of IBM Storage Scale* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Preparing the remote key management (RKM) server

The preparation of the RKM server depends on the RKM server product that you select and the encryption setup that you plan to follow. For more information, see the help topic in the following list that describes the setup of your RKM server:

- [“Simplified setup: Using SKLM with a self-signed certificate” on page 747](#)
- [“Setup using HashiCorp Vault KMIP Secrets Engine” on page 757](#)
- [“Regular setup: Using SKLM with a self-signed certificate” on page 788](#)
- [“Configuring encryption with the Thales Vormetric DSM key server” on page 812](#)

RKM back ends

An RKM back end defines a connection between a local key client, a remote key tenant, and an RKM server. Each RKM back end is described in an RKM stanza in an RKM.conf file on each node that is configured for encryption.

By controlling the contents of the RKM.conf file, the cluster administrator can control which client nodes have access to MEKs. For example, the same RKM server can be given two different names in /var/mmfs/etc/RKM.conf stanzas. Then, the administrator can partition a set of MEKs hosted on a single RKM server into separate subsets of MEKs. These subsets of MEKs might belong to subsets of the nodes of the cluster.

Because the MEKs are cached in memory, some short-term outages while a key client is attempting to access a key server might not cause issues. However, failure to retrieve the keys might result in errors while the file system is creating, opening, reading, or writing files. Although the keys are cached, they are periodically retrieved from the key server to ensure their validity.

To ensure that MEKs are always available, it is a good practice to set up multiple key servers in a high-availability configuration. See the subtopic [“Adding backup RKM servers in a high-availability configuration” on page 746](#).

The RKM.conf file and the RKM stanza

The management of the RKM.conf file and its stanzas depends on the setup:

- In the simplified setup, the **mmkeyserv** command manages its own RKM.conf file and updates it automatically. This management includes adding any backup servers for High Availability and other key retrieval properties.
- In the regular setup and the DSM setup, you must manage the RKM.conf file and its contents.

The location of the RKM.conf file also depends on the setup:

Table 57. The RKM.conf file	
Setup	Location of the RKM.conf file
Simplified setup	/var/mmfs/ssl/keyServ/RKM.conf
Regular setup	/var/mmfs/etc/RKM.conf
DSM setup	/var/mmfs/etc/RKM.conf

The length of the RKM.conf file cannot exceed 1 MiB. No limit is set on the number of RKM stanzas, if the length limit is not exceeded.

After the file system is configured with encryption policy rules, the file system is considered encrypted. From that point on, each node that has access to that file system must have an RKM.conf file present. Otherwise, the file system might not be mounted or might become unmounted.

Each RKM stanza in the RKM.conf file describes a connection between a local key client, a remote tenant, and an RKM server. The following code block shows the structure of an RKM stanza:

```
RKM ID {
    type = ISKLM
    kmipServerUri = tls://host:port
    keyStore = PathToKeyStoreFile
    passphrase = Password
    clientCertLabel = LabelName
    tenantName = NameOfTenant
    [connectionTimeout = ConnectionTimeout]
    [connectionAttempts = ConnectionAttempts]
    [retrySleep = RetrySleepUsec]
}
```

The following list describes the terms of the stanza:

RKM ID

The name of the stanza. The RKM ID must be unique within the cluster, must be 1 - 21 characters in length, and can contain only the characters a - z, A - Z, 0 - 9, and underscore (_). The first character cannot be a numeral.

type

ISKLM for the regular setup and the simplified setup of GKLM. KMIP for the DSM setup, CipherTrust Manager setup, and HashiCorp Vault KMIP Secrets Engine setup.

kmipServerUri

The DNS name or IP address of the SKLM or DSM server and the KMIP SSL port.

keyStore

The path and name of the client keystore.

passphrase

The password of the client keystore and client certificate.

clientCertLabel

The label of the client certificate in the client keystore.

tenantName

The name of the tenant or device group.

connectionTimeout

The connection timeout, in seconds. The default is 60 seconds. The valid range is 1 - 120 seconds.

connectionAttempts

The number of connection attempts. The default is three attempts. The valid range is 1 - 10.

retrySleep

The retry sleep time, in microseconds. The default is 100,000 (0.1 seconds). The valid range is 1 - 10,000,000 microseconds.

The client keystore directory and its files

The files in the client keystore directory include the client keystore file, the public and private key files for the client, and possibly other files that are described in later topics.

The management of these files depends on the setup:

- In the simplified setup, the **mmkeyserv** command creates and updates the files in the client keystore directory automatically.
- In the regular setup and the DSM setup, you must run various commands to create and update the files.

The location of the client keystore directory also depends on the setup:

Table 58. The client keystore directory	
Setup	Location of the client keystore directory
Simplified setup	/var/mmfs/ssl/keyServ
Regular setup	/var/mmfs/etc/RKMcerts
DSM setup	/var/mmfs/etc/RKMcerts

Adding backup RKM servers in a high-availability configuration

You can add up to five backup RKM servers to your configuration if necessary to improve the reliability or performance of master encryption key retrieval. A backup RKM server is specified by adding a line in the following format to the RKM stanza:

```
<server_name>=<IP_address:port_number>
```

The line must be added either immediately after the line that specifies the primary RKM server or immediately after a line that specifies another backup RKM server. In the following example, the maximum of five backup RKM servers is specified:

```
rkmname3 {  
    type = ISKLM  
    kmipServerUri = tls://host:port  
    kmipServerUri2 = tls://host:port # TLS connection to backup RKM server 1  
    kmipServerUri3 = tls://host:port # TLS connection to backup RKM server 2  
    kmipServerUri4 = tls://host:port # TLS connection to backup RKM server 3  
    kmipServerUri5 = tls://host:port # TLS connection to backup RKM server 4  
    kmipServerUri6 = tls://host:port # TLS connection to backup RKM server 5  
    keyStore = PathToKeyStoreFile  
    passphrase = Password  
    clientCertLabel = LabelName  
    tenantName = NameOfTenant  
    [connectionTimeout = ConnectionTimeout]  
    [connectionAttempts = ConnectionAttempts]  
    [retrySleep = RetrySleepUseC]  
}
```

Note: In the regular setup method, you must add each line manually; in the simplified setup lines are added automatically in response to **mmkeyserv** commands. For more information, see the following subtopics:

- Regular setup: See the subtopic "Part 3: Configuring the remote key management (RKM) back end" in the topic ["Regular setup: Using SKLM with a self-signed certificate" on page 788](#).

- Simplified setup: See the subtopic "Adding backup key servers" in the topic [“Simplified setup: Doing other tasks” on page 780](#).

If at least one backup RKM server is configured, then whenever key retrieval from the primary RKM server fails, IBM Storage Scale queries each backup RKM server in the list, in order, until it finds the MEK. The addition of the URIs for the backup RKM servers is the only change that is required within IBM Storage Scale. All other configuration parameters (certificates, keys, node, and tenant information) do not need to change, because they are also part of the set of information that is replicated. The administrator is responsible for creating and maintaining any backups.

Additionally, setting up backup RKM servers can help gain some performance advantage by distributing MEK retrieval requests across the backup RKM servers in a round-robin fashion. To achieve this result, the administrator must specify different orderings of the server endpoints on different IBM Storage Scale nodes in the `/var/mmfs/etc/RKM.conf` file.

Note: The primary and all secondary RKM servers in high-availability setup must use the same KMIP certificate.

For example, if two backup RKM servers are available, such as `tls://keysrv.ibm.com:5696` and `tls://keysrv_backup.ibm.com:5696`, half of the nodes in the cluster can have the following content in `/var/mmfs/etc/RKM.conf`:

```
...
kmipServerUri = tls://keysrv.ibm.com:5696
kmipServerUri2 = tls://keysrv_backup.ibm.com:5696
...
```

The other half can use the following content:

```
...
kmipServerUri = tls://keysrv_backup.ibm.com:5696
kmipServerUri2 = tls://keysrv.ibm.com:5696
...
```

Establishing an encryption-enabled environment

The steps for establishing an encryption-ready environment depend on the version of IBM Storage Scale and on the type and version of the Remote Key Manager (RKM) server.

Each of the following subtopics topics describes how to configure a basic setup with a single encrypted fileset. Three deployment scenarios are supported:

- IBM Storage Scale 4.2 or later and the simplified setup method.
- IBM Storage Scale 4.2 or later and a supported version of Thales Vormetric Data Security Manager (DSM). For more information, see [“Preparation for encryption” on page 742](#).
- GPFS Advanced Edition 4.2 or later and the regular setup method.

Note: IBM Storage Scale supports IBM Security Guardium Key Lifecycle Manager (GKLM) 4.1.0.1 (IF01), 4.1.1, or later. The older versions of GKLM are referred to as IBM Security Lifecycle Manager or SKLM in the documentation. The configuration information is the same for both GKLM and SKLM.

For more information, see [“Preparation for encryption” on page 742](#).

Simplified setup: Using SKLM with a self-signed certificate

Learn how to configure IBM Security Key Lifecycle Manager (SKLM) in the simplified setup when you use a self-signed server certificate rather than a certificate chain from a certificate authority (CA).

This topic describes the simplified method for setting up encryption with SKLM as the key server and with a self-signed certificate on the KMIP port of the RKM server. For more information about the simplified setup, see the topic [“Preparation for encryption” on page 742](#).

Note: IBM Storage Scale supports IBM Security Guardium Key Lifecycle Manager (GKLM) 4.1.0.1 (IF01), 4.1.1, or later. The older versions of GKLM are referred to as IBM Security Lifecycle Manager or SKLM in the documentation. The configuration information is the same for both GKLM and SKLM.

If your deployment scenario uses a chain of certificates from a certificate authority rather than a self-signed certificate, see one of the following topics:

[“Simplified setup: Using SKLM with a certificate chain” on page 761](#)

[“Regular setup: Using SKLM with a certificate chain” on page 797](#)

The simplified setup with SKLM requires IBM Storage Scale Advanced Edition, IBM Storage Scale Data Management Edition, or IBM Storage Scale Developer Edition or IBM Storage Scale Erasure Code Edition 4.2.1 or later and a supported version of SKLM. For more information, see [“Preparation for encryption” on page 742](#).

Note: If you are using SKLM 2.7 or later, see the topic [“Configuring encryption with SKLM 2.7 or later” on page 810](#).

Requirements:

The following requirements must be met on every IBM Storage Scale node that participates in encryption:

- The node must have direct network access to the system where the key server is installed.
- The security-sensitive files that are created during the configuration process must have the following characteristics:
 - They must be regular files that are owned by the root user.
 - The group ownership must be changed to root group.
 - They must be readable and writable only by the user (mode '0600'). See the following examples:

```
-rw----- 1 root root 2454 Mar 20 10:32 /var/mmfs/ssl/keyServ/RKM.conf  
drw----- 2 root root 4096 Mar 20 11:15 /var/mmfs/ssl/keyServ/  
-rw----- 1 root root 3988 Mar 20 11:15 /var/mmfs/ssl/keyServ/keystore_name.p12
```

Note: In the simplified setup, the **mmkeyserv** command sets the permission bits automatically.

These security-sensitive files include the following files:

- The RKM.conf file. For more information about this file, see [“The RKM.conf file and the RKM stanza” on page 744](#).
- The files in the client keystore directory, which include the keystore file, the public and private key files for the client, and possibly other files. For more information about these files, see [“The client keystore directory and its files” on page 746](#).

Note: In the simplified setup, the **mmkeyserv** command automatically creates and distributes the RKM.conf files and the files in the client keystore directory to every node in the cluster. The files are located in the following directory on each node:

```
/var/mmfs/ssl/keyServ
```



CAUTION:

- Take appropriate precautions to ensure that the security-sensitive files are not lost or corrupted. IBM Storage Scale does not manage or replicate the files.
- Ensure that the passphrase for the client certificate file is not leaked through other means, such as the shell history.
- Client keystore files must be record-locked when the GPFS daemon starts. If the keystore files are stored on an NFS mount, the encryption initialization process can hang. The cause is a bug that affects the way NFS handles record locking. If you encounter this problem, upgrade your version of NFS or store your keystore file on a local file system. If an upgrade is not possible and no local file system is available, use a RAM drive to store the keystore files.

The setup procedure is greatly simplified by the use of the **mmkeyserv** command, which automates many of the tasks that must be done manually in the regular setup:

- Creating and configuring client credentials.
- Creating a device group and master encryption keys in the RKM server.
- Creating and updating RKM.conf configuration files.
- Retrieving server certificates from the RKM server and storing them in client keystores.
- Propagating configuration information and client credentials to every node in the cluster.

See the following subtopics for instructions:

[“Part 1: Installing and configuring SKLM” on page 749](#)

[“Part 2: Configuring the cluster for encryption” on page 751](#)

[“Part 3: Adding a node to the cluster” on page 757](#)

Part 1: Installing and configuring SKLM

Follow the instructions in this subtopic to install and configure SKLM on the RKM server.

1. Install SKLM. For more information, see [“Preparation for encryption” on page 742](#).

For information about installing SKLM, see the *Installing* chapter of the SKLM documentation.

2. From the main page of the SKLM web GUI, click **Configuration > Key Serving Parameters** and select the check box for **Keep pending client device communication certificates**.

3. Configure SKLM to have the same FIPS 140-2 (FIPS) setting as the IBM Storage Scale cluster.

Follow these steps:

- a) Determine the FIPS setting of the cluster by issuing the following command:

```
mmIsconfig FIPS1402mode
```

The command returns yes if the cluster complies with FIPS or no if not.

- b) On the SKLM server system, open the SKLMConfig.properties file.

Note: The default location of the SKLMConfig.properties file depends on the operating system:

- On AIX, Linux, and similar operating systems the directory is at the following location:

- For GKLM 4.1.1 or later versions:

/opt/IBM/WebSphere/Liberty/products/sklm/config/SKLMConfig.properties

- For GKLM 4.1.0.1 and old supported SKLM versions:

/opt/IBM/WebSphere/AppServer/products/sklm/config/
SKLMConfig.properties

- On Microsoft Windows, the directory is at the following location:

- For GKLM 4.1.1 or later versions:

Drive:\Program Files
(x86)\IBM\WebSphere\Liberty\products\sklm\config\SKLMConfig.properties

- For GKLM 4.1.0.1 and old supported SKLM versions:

Drive:\Program Files
(x86)\IBM\WebSphere\AppServer\products\sklm\config\SKLMConfig.properties

- c) Find the following line in the SKLMConfig.properties file:

```
fips=on
```

or

```
fips=off
```

If the line is not present in the file, then add it. Set the value to on to configure SKLM to comply with FIPS, or set it to off to configure SKLM not to comply with FIPS.

4. Configure the SKLM server to have the same NIST SP800-131a (NIST) setting as the IBM Storage Scale cluster. Follow these steps:

- Determine the NIST setting of the cluster by issuing the following command:

```
mmlsconfig nistCompliance
```

The command returns SP800-131A if the cluster complies with NIST or off if not.

- On the SKLM server system, open the `SKLMConfig.properties` file. For the location of this file, see the note in Step 3.

- Find the line in the `SKLMConfig.properties` file that begins with the following phrase:

```
TransportListener.ssl.protocols=
```

If the line is not present in the file, then add it. To configure SKLM to comply with NIST, set the value as it is shown below:

```
TransportListener.ssl.protocols=TLSv1.2
```

To configure SKLM not to comply with NIST, set the value as it is shown below:

```
TransportListener.ssl.protocols=SSL_TLS
```

5. Configure IBM WebSphere® Application Server so that it has the same NIST setting as the IBM Storage Scale cluster.

See the topic [Transitioning WebSphere Application Server to the SP800-131 security standard](#) in the volume *WebSphere Application Server Network Deployment* in the WebSphere Application Server online documentation.

- WebSphere Application Server can be configured to run SP800-131 in a transition mode or a strict mode. The strict mode is recommended.
 - When NIST is enabled, make sure that WebSphere Application Server certificate size is at least 2048 bytes and is signed with SHA256withRSA as described in the preceding link.
6. If the cipher suites were set at any time, SKLM 2.6.0.0 has a known issue that causes server certificates always to be signed with SHA1withRSA. To work around the problem, follow these steps:
 - While the SKLM server is running, in the `SKLMConfig.properties` file, modify the `requireSHA2Signatures` property as follows:

```
requireSHA2Signatures=true
```

- Do not restart the server.
 - Generate a new server certificate and set it to be the one in use.
 - If you restart the server, you must repeat this workaround before you can create a server certificate that is signed other than with SHA1withRSA.
7. Create a self-signed server certificate:
 - On the system where SKLM is running, open the graphical user interface.
 - Click **Configuration > SSL/KMIP**.
 - Click **Create self-signed certificate**.
 - Enter the information for the certificate and click **OK**.
 - Restart the server to verify that the server can operate with the new certificate.

Part 2: Configuring the cluster for encryption

Gather the following information:

- The logon password of the SKLMAdmin administrator
- The certificate chain of the SKLM server (optional)

The following table provides a high-level overview of the configuration process. The steps in the table correspond to the steps in the procedure that begins immediately after the table.

Table 59. Configuring the cluster for encryption in the simplified setup	
Step	Actions
1	Verify the direct network connection between the IBM Storage Scale node and the SKLM server.
2	Add the SKLM key server to the configuration.
3	Add a tenant to the key server.
4	Create a key client.
5	Register the key client to the tenant.
6	Create a master encryption key in the tenant.
7	Set up an encryption policy in the cluster.
8	Test the encryption policy.

1. Verify that the IBM Storage Scale node that you are working from has a direct network connection to the RKM server.
2. Add the RKM server to the encryption configuration:
 - a) Use the **mmkeyserv server add** command to add the SKLM server to the encryption configuration. Depending on how SKLM is configured, you might also need to specify a port number for connecting with SKLM:
 - If SKLM is configured to use its default REST port for communications with its clients, you do not need to specify a port number when you add the server. Issue a command like the following one:

```
mmkeyserv server add ServerName
```

where:

- *ServerName* is the host name or IP address of the SKLM key server that you want to add.

When no port number is specified, IBM Storage Scale automatically tries to connect with SKLM through the default REST port number of each of the supported versions of SKLM serially, starting with the earliest version, until it finds a successful connection with SKLM.

Note: The default REST port number depends on the version of SKLM that is installed on the RKM server. For more information, see *Firewall recommendations for IBM SKLM* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

- If SKLM is not configured to use its default REST port number, you must specify the correct port number when you add the server. Issue a command like the following one:

```
mmkeyserv server add ServerName --port RestPortNumber
```

where:

- *ServerName* is the host name or IP address of the SKLM key server that you want to add.
- *RestPortNumber* is the port number that SKLM uses to connect with its clients.

If you do not specify a port number or if you specify an incorrect port number, IBM Storage Scale fails to connect with SKLM and displays an error message. For more information see the description of the --port parameter in **mmkeyserv command** in the *IBM Storage Scale: Command and Programming Reference Guide*.

- b) Respond to the prompts from the **mmkeyserv server add** command. See the example output and prompts in the figure that follows:
 - i) Enter the SKLM administrator password when prompted.
 - ii) To view the certificate chain of the SKLM server, enter **view** when prompted.
 - iii) Verify that the certificates that are displayed have the same contents as the certificates in the chain that you downloaded from SKLM.
 - iv) Enter **yes** to trust the certificates or **no** to reject them.
 - v) If you trust the certificates, the command adds the RKM server to the encryption configuration.
-
- In the following listing, key server keyserver01 is added:

```
# mmkeyserv server add keyserver01
Enter password for the key server keyserver01:
The security certificate(s) from keyserver01.gpfs.net must be accepted to continue. View the
certificate(s) to determine whether you want to trust the certifying authority.
Do you want to view or trust the certificate(s)? (view/yes/no) view

Serial number: 01022a8adf20f3
SHA-256 digest: 2ca4a48a3038f37d430162be8827d91eb584e98f5b3809047ef4a1c72e15fc4c
Signature: 7f0312e7be18efd72c9d8f37dbb832724859ba4bb5827c230e2161473e0753b367ed49d
993505bd23858541475de8e021e0930725abbd3d25b71edc8fc3de20b7c2db5cd4e865f41c7c410c1d710acf222e1c4
5189108e40568ddcbeb21094264da60a1d96711015a7951eb2655363309d790ab44ee7b26adf8385e2c210b8268c5ae
de5f82f268554a6fc22ece6feeee2a6264706e71416a0dbe8c39ceacd86054d7cc34dda4fffea4605c037d321290556
10821af85dd9819a4d7e4baa70c51addcdca720d33bc9f8bbde6d292c028b2f525a0275ebea968c26f8f0c4b604719ae
3b04e71ed7a818cd6adf68764374b29c91df3d101a941bf8b7189485ad72
Signature algorithm: SHA256WithRSASignature
Key size: 2048
Issuer: C=US, O=IBM, OU=SKLMNode, SKLMCell, Root Certificate, CN=c40bbc1xn3.gpfs.net
Subject: C=US, O=IBM, OU=SKLMNode, SKLMCell, CN=c40bbc1xn3.gpfs.net

Serial number: 01022a24475466
SHA-256 digest: 077c3b53c5046aa893b760c11cca3a993efbc729479771e03791f9ed4f716879
Signature: 227b5bef89f2e55ef628da6b50db1ab842095a54e1505655e3d95fee753a7f7554868a
a79b294c503dc34562cf69c2a20128796758838968565c0812c4aedd0543d396646a269c02bf4c5ce5acba4409a10e
ffbd47ca38ce492698e2dc8d390b9ae3f4a47c23ee3045ff0145218668f35a63edac68201789ed0db6e5c170f5c6db
49769f0b4c9a5f208746e4342294c447793ed087fa0ac762588faf420febeb3fca411e4e725bd46476e1f9f44759a69
6573af5dbbc9553218c7083c208440f2e542bf56cc5cc18156cce05efd6c2e5fea2b886c5c1e262c10af18b13ccf38c3
533ba025b97bbe62f271545b2ab5c1f50c1dca45ce504dfcf257362e9b43
Signature algorithm: SHA256WithRSASignature
Key size: 2048
Issuer: C=US, O=IBM, OU=SKLMNode, SKLMCell, Root Certificate, CN=c40bbc1xn3.gpfs.net
Subject: C=US, O=IBM, OU=SKLMNode, SKLMCell, Root Certificate, CN=c40bbc1xn3.gpfs.net

Do you trust the certificate(s) above? (yes/no) yes
```

Figure 27. Example listing for **mmkeyserv server add**

-
- c) Issue the **mmkeyserv server show** command to verify that the key server is added. The following listing shows that keyserver01 is created:

```
# mmkeyserv server show
keyserver01
  Type: ISKLM
  Hostname: keyserver01.gpfs.net
  User ID: SKLMAdmin
  REST port: 9080
  Label: 1_keyserver01
  NIST: on
  FIPS1402: off
  Backup Key Servers:
  Distribute: yes
  Retrieval Timeout: 120
  Retrieval Retry: 3
  Retrieval Interval: 10000
```

```
REST Certificate Expiration: 2033-05-18 17:01:24 (-0400)
KMIP Certificate Expiration: 2021-05-22 22:24:54 (-0400)
```

3. Issue the **mmkeyserv tenant add** command to add a tenant to the key server. The command creates the tenant on the SKLM server if it does not exist.

A *tenant* is an entity on the SKLM server that can contain encryption keys and certificates. SKLM uses the term *device group* instead of *tenant*.

- a) Issue the following command to add tenant devG1 to key server keyserver01. Enter the SKLM administrator password when prompted:

```
# mmkeyserv tenant add devG1 --server keyserver01
Enter password for the key server keyserver01:
```

- b) Issue the **mmkeyserv tenant show** command to verify that the tenant is added. The following listing shows that tenant devG1 is added to keyserver01:

```
# mmkeyserv tenant show
devG1
  Key Server:      keyserver01.gpfs.net
  Registered Client: (none)
```

4. Issue the **mmkeyserv client create** command to create a key client. A key client can request master encryption keys from a tenant after it is registered to the tenant. The command creates a client keystore on the node from which the command is issued and puts into the keystore a set of client credentials and the certificate chain of the SKLM server. The command then copies the keystore to all the nodes in the cluster. The keystore is stored in the following directory on each node of the cluster:

```
/var/mmfs/ssl/keyServ
```

- a) Issue the following command to create key client c1Client1 for key server keyserver01. Enter the SKLM administrator password and a passphrase for the new keystore when prompted:

```
# mmkeyserv client create c1Client1 --server keyserver01
Enter password for the key server keyserver01:
Create a pass phrase for keystore:
Confirm your pass phrase:
```

Alternatively, issue the following command to create key client c1Client1 for key server keyserver01 using a user-provided, CA-signed certificate. The client certificate file is `client1CertFile.cert`, the client's key file is `client1PrivFile.pem`, and the CA chain file is `CACertChain.pem`. Enter the SKLM administrator password and a passphrase for the new keystore when prompted:

```
# mmkeyserv client create c1Client1 --server keyserver01 -cert client1CertFile.cert
  --priv client1PrivFile.pem --ca-chain CACertChain.pem
Enter password for the key server keyserver01:
Create a pass phrase for keystore:
Confirm your pass phrase:
```

There are three elements to using external certificates:

- A CA-signed certificate file, which certifies the client's identity.
- A private key file that matches the client's certificate.
- The certificate chain of the CA that signed the client certificate.

All these elements must be provided to the **mmkeyserv** command to establish trust in the client's identity and to use it to create a secure connection with the SKLM server. The certificates must be in PEM-encoded x509 format, and the content of the private key file must be PEM-encoded and unencrypted.

The CA certificate chain can be used either as individual files, one file for each CA certificate in the chain, or as a chain file that contains all the CA certificates:

- To create a chain file, concatenate all the CA certificates from the certificate authority into a single file. The file must begin with the CA root certificate, continue with the intermediate CA

certificates in the order in which they are used, and end with the CA certificate that signed the client certificate.

- To use the CA certificates as individual files, copy them to a temporary location and rename each file using the format <CACertFilesPrefix>. <n>. cert, where <CACertFilesPrefix> is the full path prefix for the CA certificate files, such as /tmp/CA/certfiles, and <n> is a CA certificate index. The index is 0 for the CA root certificate and n - 1 for the last intermediate CA certificate that signed the client certificate.

In the following example, the chain consists of a CA root certificate file and two intermediate CA certificate files. The full path prefix is /tmp/CA/certfiles:

CA root certificate	/tmp/CA/certfiles.0.cert
First intermediate CA root certificate:	/tmp/CA/certfiles.1.cert
Second intermediate CA root certificate:	/tmp/CA/certfiles.2.cert

Issue the following command to create key client c1Client1 for key server keyserver01:

```
# mmkeyserv client create c1Client1 --server keyserver01 --cert client1CertFile.cert --priv client1PrivFile.pem --ca-cert /tmp/CA/certfiles  
Enter password for the key server keyserver01:  
Create a pass phrase for keystore:  
Confirm your pass phrase:
```

- b) Issue the **mmkeyserv client show** command to verify that the key client is created. The Certificate Type attribute is set to user-provided if the client was created with a CA-signed certificate or to system-generated if the client was created with a self-signed certificate that was generated by IBM Storage Scale.

In the following example, the output shows that key client c1Client1 was created for remote key server keyserver01.gpfs.net and that the client certificate is a system-generated, self-signed certificate:

```
# mmkeyserv client show  
c1Client1  
Label: c1Client1  
Key Server: keyserver01.gpfs.net  
Tenants: (none)  
Certificate Expiration: 2023-03-11 00:01:03 (-0500)  
Certificate Type: system-generated
```

In the following example, the output shows that key client c1Client1 was created with a user-provided, CA-signed certificate:

```
# mmkeyserv client show  
c1Client1  
Label: c1Client1  
Key Server: keyserver01.gpfs.net  
Tenants: (none)  
Certificate Expiration: 2023-03-11 00:01:03 (-0500)  
Certificate Type: user-provided
```

5. Issue the **mmkeyserv client register** command to register the key client with the tenant:

You must provide a remote key management (RKM) ID as an input for this command. The RKM ID will become the identifier field of a new RKM stanza that describes the connection between this key client, this tenant, and this key server. For more information about the RKM stanza, see “[The RKM.conf file and the RKM stanza](#)” on page 744.

It is a good practice to use a format like the following one to ensure that the RKM ID is unique:

```
keyServerName_tenantName
```

For example, the RKM ID for the key server and the tenant in these instructions is keyserver01_devG1.

- a) Issue the following command to register key client c1Client1 with tenant devG1 under RKM ID keyserver01_devG1. Enter the requested information when prompted:

```
# mmkeyserv client register c1Client1 --tenant devG1 --rkm-id keyserver01_devG1
Enter password for the key server:
mmkeyserv: [I] Client currently does not have access to the key. Continue the
registration
process ...
mmkeyserv: Successfully accepted client certificate
```

- b) Issue the command **mmkeyserv tenant show** to verify that the key client is known to the tenant.

The following listing shows that tenant devG1 lists c1Client1 as a registered client:

```
mmkeyserv tenant show
devG1
Key Server:      keyserver01.gpfs.net
Registered Client:  c1Client1
```

- c) You can also issue the command **mmkeyserv client show** to verify that the tenant is known to the client.

The following listing shows that client c1Client1 is registered with tenant devG1:

```
# mmkeyserv client show
c1Client1
Label:          c1Client1
Key Server:    keyserver01.gpfs.net
Tenants:        devG1
Certificate Expiration: 2023-03-11 00:01:03 (-0500)
```

- d) To see the contents of the RKM stanza, issue the **mmkeyserv rkm show** command.

In the following listing, notice that the RKM ID of the stanza is keyserver01_devG1, the string that was specified in Step 5(a):

```
# mmkeyserv rkm show
keyserver01_devG1 {
type = ISKLM
kmipServerUri = tls://192.0.2.59:5696
keyStore = /var/mmfs/ssl/keyServ/serverKmip.1_keyserver01.c1Client1.1.p12
passphrase = pw4c1Client1
clientCertLabel = c1Client1
tenantName = devG1
}
```

- e) You can also see the RKM stanza by displaying the contents of the RKM.conf file on the node:

```
# cat /var/mmfs/ssl/keyServ/RKM.conf
keyserver01_devG1 {
  type = ISKLM
  kmipServerUri = tls://192.0.2.59:5696
  keyStore = /var/mmfs/ssl/keyServ/serverKmip.1_keyserver01.c1Client1.1.p12
  passphrase = pw4c1Client1
  clientCertLabel = c1Client1
  tenantName = devG1
}
```

6. Issue the **mmkeyserv key create** command to create a master encryption key in the tenant. The following command creates a master encryption key in tenant devG1 of server keyserver01.gpfs.net.

The command displays the UUID of the encryption key (not the key value itself) at line 3 of the listing:

```
# mmkeyserv key create --server keyserver01.gpfs.net --tenant devG1
Enter password for the key server keyserver01.gpfs.net:
KEY-d4e83148-e827-4f54-8e5b-5e1b5cc66de1
```

7. Set up an encryption policy on the node.

- a) Create a file management policy that instructs GPFS to do the encryption tasks that you want.

The following example policy instructs IBM Storage Scale to encrypt all files in the file system with a file encryption key (FEK) and to wrap the FEK with a master encryption key (MEK):

```
RULE 'p1' SET POOL 'system' /* one placement rule is required at all times */
RULE 'Encrypt all files in file system with rule E1'
SET ENCRYPTION 'E1'
WHERE NAME LIKE '%'
RULE 'simpleEncRule' ENCRYPTION 'E1' IS
ALGO 'DEFAULTNISTSP800131A'
KEYS ('KEY-d4e83148-e827-4f54-8e5b-5e1b5cc66de1:keyserver01_devG1')
```

In the last line of the policy, the character string within single quotation marks ('') is the key name. A *key name* is a compound of two parts in the following format:

KeyID:*RkmID*

where:

KeyID

Specifies the UUID of the key that you created in Step 6.

RkmID

Specifies the RKM ID that you specified in Step 5(a).

- b) Issue the **mmchpolicy** command to install the rule.



CAUTION: Installing a new policy with the **mmchpolicy** command removes all the statements in the previous policy. To add statements to an existing policy without deleting the previous contents, collect all policy statements for the file system into one file. Add the new statements to the file and install the contents of the file with the **mmchpolicy** command.

- i) Issue the following command to install the policy rules in file enc.pol for file system c1FileSystem1:

```
# mmchpolicy c1FileSystem1 /tmp/enc.pol
Validated policy 'enc.pol': Parsed 3 policy rules.
Policy 'enc.pol' installed and broadcast to all nodes.
```

- ii) You can list the new encryption policy with the following command:

```
# mm1spolicy c1FileSystem1 -L
```

8. Test the new encryption policy:

- a) Create a file in the file system c1FileSystem1:

```
# echo 'Hello World!' >/c1FileSystem1/hw.enc
```

The policy engine detects the new file, encrypts it, and wraps the file encryption key in a master encryption key.

- b) To verify that the file hw.enc is encrypted, issue the following command to display the encryption attribute of the file.

The output shows that the file is encrypted:

```
# mm1sattr -n gpfs.Encryption /c1Filesystem1/hw.enc
file name:          /c1Filesystem1/hw.enc
gpfs.Encryption:   "EAGC????.????????????? ?????h????????????????? ?u?~?}?????????????
t??1N??
'k???*?3??C??#?)?KEY-ef07b465-cfa5-4476-9f63-544e4b3cc119?NewGlobal11?"
EncPar 'AES:256:XTS:FEK:HMACSHA512'
      type: wrapped FEK WrpPar 'AES:KWRAP' CmbPar 'XORHMACSHA512'
      KEY-d4e83148-e827-4f54-8e5b-5e1b5cc66de1:keyserver01_devG1
```

Part 3: Adding a node to the cluster

- When you add a node to a cluster that is configured for encryption by the simplified setup, the cluster automatically detects the new node and copies the encryption configuration to it. For other requirements, see the Requirements section earlier in this topic.

Setup using HashiCorp Vault KMIP Secrets Engine

This topic describes how to setup encryption using HashiCorp Vault KMIP Secrets Engine as the RKM key server.

Requirements:

The following requirements must be met on every IBM Storage Scale node that participates in encryption:

- The node must have direct network access to the system where the key server is installed.
- The security-sensitive files that are created during the configuration process must have the following characteristics: The security-sensitive files include the following files:
 - The RKM.conf file. For more information about this file, see [“The RKM.conf file and the RKM stanza” on page 744](#).
 - The files in the client keystore directory, which include the keystore file, the public and private key files for the client, and possibly other files. For more information about these files, see [“The client keystore directory and its files” on page 746](#).

Note: In the simplified setup, the **mmkeyserv** command automatically creates and distributes the RKM.conf files and the files in the client keystore directory to every node in the cluster. The files are located in the following directory on each node:

```
/var/mmfs/ss1/keyServ
```

- Client keystore files must be record-locked when the GPFS daemon starts. If the keystore files are stored on an NFS mount, the encryption initialization process can hang. The cause is a bug that affects the way NFS handles record locking. If you encounter this problem, upgrade your version of NFS or store your keystore file on a local file system. If an upgrade is not possible and no local file system is available, use a RAM drive to store the keystore files.

See the following subtopics for instructions:

[Part 1: Installing and configuring Vault KMIP Secrets Engine](#)

[“Part 2: Configuring IBM Storage Scale cluster to use HashiCorp Vault KMIP Secrets Engine for encryption” on page 757](#)

Part 1: Installing and configuring Vault KMIP Secrets Engine

Follow the below links to install and configure HashiCorp Vault KMIP Secrets Engine:

- [Vault Documentation](#)
- [Installing Vault](#)

Part 2: Configuring IBM Storage Scale cluster to use HashiCorp Vault KMIP Secrets Engine for encryption

Gather the following information:

- The Vault Enterprise key server hostname.
- Create and save a temporarily authentication token in a file with owner only read permission.

The following table provides a high-level overview of the configuration process. The steps in the table correspond to the steps in the procedure that begins immediately after the table.

Table 60. Configuring the cluster to use HashiCorp Vault KMIP Secrets Engine

Step	Actions
1	Verify the direct network connection between the IBM Storage Scale node and the RKM server.
2	Add the RKM key server to the configuration.
3	Create a new role from a scope on the RKM server.
4	Register the role created in the previous step.
5	Create a master encryption key for the role.
6	Set up an encryption policy in the node.
7	Test the encryption policy.

1. Verify that the IBM Storage Scale node that you are working from has a direct network connection to the RKM server.
2. Add the RKM server to the encryption configuration:

a) Use the **mmkeyserv server add** command to add the RKM server.

For example:

```
# mmkeyserv server add tru-4pub.fyre.ibm.com --auth-token tempToken
mmkeyserv: mmsdrfs propagation completed.
```

In this example:

- *tru-4pub.fyre.ibm.com* is the host name of the Vault Enterprise key server.
- *tempToken* contains a temporarily token that given by the Vault administrator.

If Vault Enterprise is not configured to use the default REST port of 8200, you must specify the **-port** option and provide the correct port number when adding the server.

HashiCorp Vault Enterprise supports high availability clusters to protect against outages. High availability mode is automatically enabled when using a data store that supports it.

HashiCorp recommends Vault Integrated Storage as the default high availability backend for new deployments of Vault. IBM Storage Scale encryption high availability with HashiCorp Vault Enterprise was tested with **Vault Integrated Storage** backend.

Configure HashiCorp Vault Enterprise high availability cluster. For more information, see [High Availability Mode \(HA\)](#). Use the **--backup** parameter and specify a comma-separated list of the standby server names. If an IBM Storage Scale node cannot retrieve a master encryption key from the primary key server, it tries the next server in the list until it either retrieves the key or exhausts the server list.

- b) Issue the **mmkeyserv server show** command to verify that the key server is added. The following listing shows that *tru-4pub.fyre.ibm.com* is added:

```
# mmkeyserv server show
tru-4pub.fyre.ibm.com
  Type:          KMIP
  IPA:          9.46.79.137
  User ID:      N/A
  REST port:    8200
  Label:        1_tru-4pub
  NIST:         on
  FIPS1402:     off
  Backup Key Servers:
  Distribute:   yes
  Retrieval Timeout: 60
  Retrieval Retry: 3
  Retrieval Interval: 10000
```

```
REST Certificate Expiration: N/A  
KMIP Certificate Expiration: N/A
```

3. Issue the **mmkeyserv role create** command to create a new role from a scope on the RKM server. The command creates the scope on the RKM server if it does not exist.

- A role name must be unique within an IBM Storage Scale cluster.
- A role name is case-insensitive and must be 1 to 16 alphanumeric characters.
- A scope name is case-insensitive and must be 1 to 16 alphanumeric characters.
- Only one role from an IBM Storage Scale cluster can be created per scope. This makes the scope unique within an IBM Storage Scale cluster.

For example:

```
# mmkeyserv role create gpfsAdmin --server tru-4pub.fyre.ibm.com --auth-token tempToken  
Create a pass phrase for keystore:  
Confirm your pass phrase:  
mmkeyserv: mmsdrfs propagation completed.
```

The above command issued without the `--scope` option. By default, it created the `gpfsAdmin` role in the `spectrumscale` scope.

Issue the **mmkeyserv role show** command to verify that the role is created. The following command listing shows the `gpfsAdmin` role is created under the `spectrumscale` scope:

```
# mmkeyserv role show  
gpfsAdmin  
    Key Server:          tru-4pub.fyre.ibm.com  
    Scope:              spectrumscale  
    Role Label:         1_gpfsAdmin  
    RKM Id:  
    CA Chain Expiration: 2032-05-11 12:40:00 (-0400)  
    Certificate Expiration: 2025-06-30 01:48:44 (-0400)  
    Certificate Serial Number: 208366370350887968995911658331713853666704552539  
    Certificate Type:    system-generated
```

4. Use the **mmkeyserv role register** command to register the role created in Step 3. You must provide a remote key management (RKM) ID as an input for this command. The RKM ID becomes the identifier field of a new RKM stanza that describes the connection between this client and the key server. For more information about the RKM stanza, see The RKM.conf file and the RKM stanza. This command creates the RKM stanza in RKM.conf file.

For example:

```
# mmkeyserv role register gpfsAdmin --rkm-id gpfsRKMstanza  
mmkeyserv: mmsdrfs propagation completed.
```

Issue the command **mmkeyserv role show** verify that the role `gpfsAdmin` is associated with the `gpfsRKMstanza` RKM ID, as shown in the following example:

```
# mmkeyserv role show  
gpfsAdmin  
    Key Server:          tru-4pub.fyre.ibm.com  
    Scope:              spectrumscale  
    Role Label:         1_gpfsAdmin  
    RKM Id:             gpfsRKMstanza  
    CA Chain Expiration: 2032-05-11 12:40:00 (-0400)  
    Certificate Expiration: 2025-06-30 01:48:44 (-0400)  
    Certificate Serial Number: 208366370350887968995911658331713853666704552539  
    Certificate Type:    system-generated
```

You can also issue the command **mmkeyserv rkm show** to verify the RKM stanza created by the **mmkeyserv role register** command.

```
# mmkeyserv rkm show  
gpfsRKMstanza {  
    type = KMIP  
    kmipServerUri = tls://9.46.79.137:5696  
    keyStore = /var/mmfs/ssl/keyServ/roleCred.1_gpfsAdmin.1.p12  
    passphrase = pass!@#ForDemo
```

```

    clientCertLabel = 1_gpfsAdmin
}

```

The same information should be available in the RKM.conf file on the node:

```

# cat /var/mmfs/ssl/keyServ/RKM.conf
gpfsRKMstanza {
    type = KMIP
    kmipServerUri = tls://9.46.79.137:5696
    keyStore = /var/mmfs/ssl/keyServ/roleCred.1_gpfsAdmin.1.p12
    passphrase = pass!@#ForDemo
    clientCertLabel = 1_gpfsAdmin
}

```

5. Issue the **mmkeyserv key create** command to create a master encryption key for the role. The following command displays the UUID of the encryption key (not the key value itself):

```
# mmkeyserv key create --role gpfsAdmin
leCTiYYs6fUPCgQsk5SHFBtTgADJHgax
```

6. Set up an encryption policy on the node.

- a) Create a file that contains the policy rules that instructs GPFS to do the encryption tasks that you want. The following example policy instructs IBM Storage Scale to encrypt all files in the file system with a file encryption key (FEK) and to wrap the FEK with a master encryption key (MEK):

```

RULE 'p1' SET POOL 'system' /* one placement rule is required at all times */
RULE 'Encrypt all files in file system with rule E1'
    SET ENCRYPTION 'E1'
    WHERE NAME LIKE '%'
RULE 'simpleEncRule' ENCRYPTION 'E1' IS
    ALGO 'DEFAULTNISTSP800131A'
    KEYS('leCTiYYs6fUPCgQsk5SHFBtTgADJHgax:gpfsRKMstanza')

```

In the last line of the policy, the character string within single quotation marks ('') is the key name. A key name is a compound of two parts in the following format:

KeyID:RkmID

where:

KeyID

Specifies the UUID of the key that you created in [Step 5](#).

RkmID

Specifies the RKM ID that you specified in [Step 4](#).

- b) Issue the **mmchpolicy** command to install the rule.

Note: Installing a new policy with the **mmchpolicy** command removes all the statements in the previous policy. To add statements to an existing policy without deleting the previous contents, collect all policy statements for the file system into one file. Add the new statements to the file and install the contents of the file with the **mmchpolicy** command.

- i) Issue the following command to install the policy rules the /tmp/fs1.pol file for the fs1 file system:

```

# mmchpolicy fs1 /tmp/fs1.pol
Validated policy 'fs1.pol': Parsed 3 policy rules.
Policy 'fs1.pol' installed and broadcast to all nodes.

```

- ii) You can list the new encryption policy with the following command:

```

# mmlspolicy fs1 -L
RULE 'p1' SET POOL 'system' /* one placement rule is required at all times */
RULE 'Encrypt all files in file system with rule E1'
    SET ENCRYPTION 'E1'
    WHERE NAME LIKE '%'
RULE 'simpleEncRule' ENCRYPTION 'E1' IS
    ALGO 'DEFAULTNISTSP800131A'
    KEYS('leCTiYYs6fUPCgQsk5SHFBtTgADJHgax:gpfsRKMstanza')

```

7. Test the new encryption policy.

- a) Create a file in the `fs1` file system.

```
# echo 'Hello World!' > /fs1/testFile
```

The policy engine detects the new file, encrypts it, and wraps the file encryption key in a master encryption key.

- b) To verify that the file `/fs1/testFile` is encrypted, issue the following command to display the encryption attribute of the file.

```
# mmlsattr -n gpfs.Encryption /fs1/testFile
```

The output shows that the file is encrypted:

```
file name:          /fs1/testFile
gpfs.Encryption:   "EAGC????*Z????????????? ??????C>yA????????????? ?2+]???ie???aq?q?
Am?W0?Q??`?8?@??t?1UN?!?leCTiYYs6fUPCgQsk5SHFBtTgADJHgax?gpfsRKMstanza?"
EncPar 'AES:256:XTS:FEK:HMACSHA512'
      type: wrapped FEK WrpPar 'AES:KWRAP' CmbPar 'XORHMACSHA512'
           leCTiYYs6fUPCgQsk5SHFBtTgADJHgax:gpfsRKMstanza
```

Simplified setup: Using SKLM with a certificate chain

Learn how to configure IBM Security Key Lifecycle Manager (SKLM) in the simplified setup when you use a certificate chain from a certificate authority rather than a self-signed server certificate.

This topic describes the simplified method for setting up encryption with SKLM as the key server and with a certificate that is signed by a certificate authority (CA) on the KMIP port of the Remote Key Management (RKM) server. For more information about the simplified setup, see the topic [“Preparation for encryption” on page 742](#).

If your deployment scenario uses a self-signed server certificate rather than a certificate chain, see one of the following topics:

[“Simplified setup: Using SKLM with a self-signed certificate” on page 747](#)

[“Regular setup: Using SKLM with a self-signed certificate” on page 788](#)

Note: IBM Storage Scale supports IBM Security Guardium Key Lifecycle Manager (GKLM) 4.1.0.1 (IF01), 4.1.1, or later. The older versions of GKLM are referred to as IBM Security Lifecycle Manager or SKLM in the documentation. The configuration information is the same for both GKLM and SKLM.

The simplified setup with SKLM requires IBM Storage Scale Advanced Edition, IBM Storage Scale Data Management Edition, or IBM Storage Scale Developer Edition or IBM Storage Scale Erasure Code Edition 4.2.1 or later and a supported version of SKLM. For more information, see [“Preparation for encryption” on page 742](#).

Note: If you are using SKLM 2.7 or later, see the topic [“Configuring encryption with SKLM 2.7 or later” on page 810](#).

Requirements:

The following requirements must be met on every IBM Storage Scale node that participates in encryption:

- The node must have direct network access to the system where the key server is installed.
- The security-sensitive files that are created during the configuration process must have the following characteristics:
 - They must be regular files that are owned by the root user.
 - The group ownership must be changed to root group.
 - They must be readable and writable only by the user (mode '0600'). See the following examples:

```
-rw----- 1 root root 2454 Mar 20 10:32 /var/mmfs/ssl/keyServ/RKM.conf
drw----- 2 root root 4096 Mar 20 11:15 /var/mmfs/ssl/keyServ/
-rw----- 1 root root 3988 Mar 20 11:15 /var/mmfs/ssl/keyServ/keystore_name.p12
```

Note: In the simplified setup, the **mmkeyserv** command sets the permission bits automatically.

These security-sensitive files include the following files:

- The RKM.conf file. For more information about this file, see “[The RKM.conf file and the RKM stanza on page 744](#)”.
- The files in the client keystore directory, which include the keystore file, the public and private key files for the client, and possibly other files. For more information about these files, see “[The client keystore directory and its files](#)” on page 746.

Note: In the simplified setup, the **mmkeyserv** command automatically creates and distributes the RKM.conf files and the files in the client keystore directory to every node in the cluster. The files are located in the following directory on each node:

```
/var/mmfs/ssl/keyServ
```



CAUTION:

- Take appropriate precautions to ensure that the security-sensitive files are not lost or corrupted. IBM Storage Scale does not manage or replicate the files.
- Ensure that the passphrase for the client certificate file is not leaked through other means, such as the shell history.
- Client keystore files must be record-locked when the GPFS daemon starts. If the keystore files are stored on an NFS mount, the encryption initialization process can hang. The cause is a bug that affects the way NFS handles record locking. If you encounter this problem, upgrade your version of NFS or store your keystore file on a local file system. If an upgrade is not possible and no local file system is available, use a RAM drive to store the keystore files.

The setup procedure is greatly simplified by the use of the **mmkeyserv** command, which automates many of the tasks that must be done manually in the regular setup:

- Creating and configuring client credentials.
- Creating a device group and master encryption keys in the RKM server.
- Creating and updating RKM.conf configuration files.
- Retrieving server certificates from the RKM server and storing them in client keystores.
- Propagating configuration information and client credentials to every node in the cluster.

See the following subtopics for instructions:

[“Part 1: Installing and configuring SKLM” on page 762](#)

[“Part 2: Configuring SKLM” on page 764](#)

[“Part 3: Configuring the cluster for encryption” on page 765](#)

[“Part 4: Adding a node to the cluster” on page 772](#)

Part 1: Installing and configuring SKLM

Follow the instructions in this subtopic to install and configure SKLM on the RKM server.

1. Install SKLM. For more information, see “[Preparation for encryption](#)” on page 742 and the [Installing and configuring](#) chapter in IBM Security Guardium® Lifecycle Manager documentation.
2. From the main page of the SKLM web GUI, click **Configuration > Key Serving Parameters** and select the check box for **Keep pending client device communication certificates**.
3. Configure SKLM to have the same FIPS 140-2 (FIPS) setting as the IBM Storage Scale cluster.

Follow these steps:

- a) Determine the FIPS setting of the cluster by issuing the following command:

```
mmlsconfig FIPS1402mode
```

The command returns yes if the cluster complies with FIPS or no if not.

b) On the SKLM server system, open the `SKLMConfig.properties` file.

Note: The default location of the `SKLMConfig.properties` file depends on the operating system:

- On AIX, Linux, and similar operating systems the directory is at the following location:

- For GKLM 4.1.1 or later versions:

`/opt/IBM/WebSphere/Liberty/products/sklm/config/SKLMConfig.properties`

- For GKLM 4.1.0.1 and old supported SKLM versions:

`/opt/IBM/WebSphere/AppServer/products/sklm/config/
SKLMConfig.properties`

- On Microsoft Windows, the directory is at the following location:

- For GKLM 4.1.1 or later versions:

`Drive:\Program Files
(x86)\IBM\WebSphere\Liberty\products\sklm\config\SKLMConfig.properties`

- For GKLM 4.1.0.1 and old supported SKLM versions:

`Drive:\Program Files
(x86)\IBM\WebSphere\AppServer\products\sklm\config\SKLMConfig.properties`

c) In the `SKLMConfig.properties` file, find the line that begins `fips=`. To configure the FIPS setting for SKLM, enter `fips=on` to comply with FIPS or `fips=off` not to comply. If the line is not present in the file, add it.

4. Configure the SKLM server to have the same NIST SP800-131a (NIST) setting as the IBM Storage Scale cluster. Follow these steps:

a) Determine the NIST setting of the cluster by issuing the following command:

```
mmlsconfig nistCompliance
```

The command returns `SP800-131A` if the cluster complies with NIST or `off` if not.

b) On the SKLM server system, open the `SKLMConfig.properties` file. For the location of this file, see the note in Step 3.

c) Add the following line to configure SKLM to comply with NIST or remove it to configure SKLM not to comply with NIST:

```
TransportListener.ssl.protocols=TLSv1.2
```

5. Configure IBM WebSphere Application Server so that it has the same NIST setting as the IBM Storage Scale cluster.

See the topic [Transitioning WebSphere Application Server to the SP800-131 security standard](#) in the volume *WebSphere Application Server Network Deployment* in the WebSphere Application Server online documentation.

- WebSphere Application Server can be configured to run SP800-131 in a transition mode or a strict mode. The strict mode is recommended.
- When NIST is enabled, make sure that WebSphere Application Server certificate size is at least 2048 bytes and is signed with SHA256withRSA as described in the preceding link.

6. If the cipher suites were set at any time, SKLM 2.6.0.0 has a known issue that causes server certificates always to be signed with SHA1withRSA. To work around the problem, follow these steps:

a) While the SKLM server is running, in the `SKLMConfig.properties` file, modify the `requireSHA2Signatures` property as follows:

```
requireSHA2Signatures=true
```

b) Do not restart the server.

- c) Generate a new server certificate signing request (CSR) to a third-party certificate authority (CA) and send it to the CA.
- d) When you receive the certificate from the third-party CA, import it into SKLM and set it to be the certificate in use. For more information, see the next subtopic.
- e) If you restart the server, you must repeat this workaround before you can create a server certificate that is signed other than with SHA1withRSA.

Part 2: Configuring SKLM

To configure SKLM, you must create a certificate signing request (CSR), send it to the certificate authority (CA), obtain the certificate chain from the CA, and import the endpoint certificate into the SKLM server.

Note: For more information about the steps in this subtopic, see [Scenario: Request for a third-party certificate in IBM Security Guardium Key Lifecycle Manager documentation](#).

1. Create a certificate signing request (CSR) with the SKLM command line interface:
 - a) On the SKLM server system, open a command line window.
 - b) Change to the *WAS_HOME/bin* directory. The location of this directory depends on the operating system:
 - On AIX, Linux, and similar operating systems, the directory is at the following location:
 - For GKLM 4.1.1 and later versions:
/opt/IBM/WebSphere/Liberty/bin
 - For GKLM 4.1.0.1 and old supported SKLM versions:
/opt/IBM/WebSphere/AppServer/bin
 - On Microsoft Windows, the directory is at the following location:
 - For GKLM 4.1.1 and later versions:
drive:\Program Files (x86)\IBM\WebSphere\Liberty\bin
 - For GKLM 4.1.0.1 and old supported SKLM versions:
drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin

- c) Start the command line interface to SKLM:

- On AIX, Linux, and similar operating systems, enter the following command:

```
./wsadmin.sh -username SKLMAAdmin -password mypwd -lang jython
```

- On Microsoft Windows, enter the following command:

```
wsadmin -username SKLMAAdmin -password mypwd -lang jython
```

- d) In the SKLM command line interface, enter the following command on one line:

```
print AdminTask.tkLMCertGenRequest('[-alias labelCsr -cn server  
-validity daysValid -keyStoreName defaultKeyStore -fileName fileName -usage SSLSERVER]')
```

where:

-alias labelCsr

Specifies the certificate label of the CSR.

-cn server

Specifies the common name of the server in the certificate.

-validity daysValid

Specifies the validity period of the certificate in days.

-keyStoreName defaultKeyStore

Specifies the keystore name within SKLM where the CSR is stored. Typically, you specify defaultKeyStore as the name here.

-fileName fileName

Specifies the fully qualified path of the directory where the CSR is stored on the SKLM server system, for example /root/sklmServer.csr.

-usage SSLSERVER

Specifies how the generated certificate is used in SKLM.

The following example shows the SKLM response:

```
CTGKM0001I Command succeeded  
fileName
```

2. Send the CSR file from Step 1 to the certificate authority.
3. When you receive the generated certificate file, or *endpoint certificate* file, from the certificate authority, copy it to a directory on the node that you are working from. For example, you might copy it to the directory and file /opt/IBM/WebSphere/Liberty/products/sklm/data/sklmServer.cert.

Important: You must also obtain and copy the root certificate file and any intermediate certificate files into the same temporary directory. The root certificate and the intermediate certificates might be included with the generated endpoint certificate file. Or you might have to obtain the root certificate file and any intermediate certificate files separately. Whatever the method, you must have a root certificate file, any intermediate certificate files, and the endpoint certificate file. You need these certificate files in Part 3.

4. Import the endpoint certificate into the SKLM server with the SKLM graphical user interface:
 - a) On the **Welcome** page, in the **Action Items** section, in the **Key Groups and Certificates** area, click **You have pending certificates**.
 - b) In the **Pending Certificates** table, click the certificate that you want to import and click **Import**.
 - c) In the **File name and location** field, type the path and file name of the certificate file and click **Import**.

Part 3: Configuring the cluster for encryption

Gather the following information:

- The logon password of the SKLMAdmin administrator
- The certificate chain of the SKLM server

The following table provides a high-level overview of the configuration process. The steps in the table correspond to the steps in the procedure that begins immediately after the table.

Table 61. Configuring the cluster for encryption in the simplified setup	
Step	Actions
1	Verify the direct network connection between the IBM Storage Scale node and the SKLM server.
2	Add the SKLM key server to the configuration.
3	Add a tenant to the key server.
4	Create a key client.
5	Register the key client to the tenant.
6	Create a master encryption key in the tenant.
7	Set up an encryption policy in the cluster.

Table 61. Configuring the cluster for encryption in the simplified setup (continued)

Step	Actions
8	Test the encryption policy.

1. Verify that the IBM Storage Scale node that you are working from has a direct network connection to the RKM server.
2. Add the RKM server to the encryption configuration:
 - a) Copy and rename the certificates:
 - i) Copy the files of the server certificate chain into a directory on the node that you are working from. A good location is the same directory in which the `keystore.pwd` file is located. Rename each certificate file with the same prefix, followed by a numeral that indicates the order of the certificate in the chain, followed by the file extension `.cert`. Start the numbering with 0 for the root certificate. For example, if the chain consists of three certificate files and the prefix is `sklmChain`, rename the files as follows:

```
sklmChain.0.cert
sklmChain.1.cert
sklmChain.2.cert
```

Note: Make sure that each certificate file contains only one certificate. In case a certificate file contains two or more CA certificates (root, intermediate, or end-point), split that file into multiple files, such as each file contains a single certificate. You must be careful about the order of certificate in the chain when you add the certificate index to the certificate file name.

- b) Use the `mmkeyserv server add` command to add the SKLM server to the encryption configuration. Depending on how SKLM is configured, you might also need to specify a port number for connecting with SKLM:
 - If SKLM is configured to use its default REST port for communications with its clients, you do not need to specify a port number when you add the server. Issue a command like the following one:

```
mmkeyserv server add ServerName --kmip-cert CertFilesPrefix
```

where:

- `ServerName` is the host name or IP address of the SKLM key server that you want to add.
- `CertFilesPrefix` is the path and the file name prefix of the files in the certificate chain. For the files from the example in the previous step, the path and file name prefix is `/root/sklmChain`. For more information, see *mmkeyserv command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

When no port number is specified, IBM Storage Scale automatically tries to connect with SKLM through the default REST port number of each of the supported versions of SKLM serially, starting with the earliest version, until it finds a successful connection with SKLM.

Note: The default REST port number depends on the version of SKLM that is installed on the RKM server. For more information, see *Firewall recommendations for IBM SKLM* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

- If SKLM is not configured to use its default REST port number, you must specify the correct port number when you add the server. Issue a command like the following one:

```
mmkeyserv server add ServerName --kmip-cert CertFilesPrefix --port RestPortNumber
```

where:

- `ServerName` is the host name or IP address of the SKLM key server that you want to add.
- `CertFilesPrefix` is the path and the file name prefix of the files in the certificate chain. For the files from the example in the previous step, the path and file name prefix is `/root/`

`sklmChain`. For more information, see *mmkeyserv command in the IBM Storage Scale: Command and Programming Reference Guide*.

- `RestPortNumber` is the port number that Security Key Lifecycle Manager uses to connect with its clients.

If you do not specify a port number or if you specify an incorrect port number, IBM Storage Scale fails to connect with SKLM and displays an error message. For more information, see *mmkeyserv command in the IBM Storage Scale: Command and Programming Reference Guide*.

- c) Respond to the prompts by the **mmkeyserv add server** command. See the example output and prompts in the figure that follows:

- i) Enter the SKLM administrator password when prompted.
- ii) To view the certificate chain of the SKLM server, enter `view` when prompted.
- iii) Verify that the certificates that are displayed have the same contents as the certificates in the chain that you downloaded from SKLM.
- iv) Enter `yes` to trust the certificates or `no` to reject them.
- v) If you trust the certificates, the command adds the RKM server to the encryption configuration.

In the following listing, key server `keyserver01` is added:

```
# mmkeyserv server add keyserver01
Enter password for the key server keyserver01:
The security certificate(s) from keyserver01.gpfs.net must be accepted to continue. View the
certificate(s) to determine whether you want to trust the certifying authority.
Do you want to view or trust the certificate(s)? (view/yes/no) view

Serial number:          01022a8adf20f3
SHA-256 digest:        2ca4a48a3038f37d430162be8827d91eb584e98f5b3809047ef4a1c72e15fc4c
Signature:              7f0312e7be18efd72c9d8f37dbb832724859ba4bb5827c230e2161473e0753b367ed49d
993505bd23858541475de8e021e0930725abbd3d25b71edc8fc3de20b7c2db5cd4e865f41c7c410c1d710acf222e1c4
5189108e40568ddcb21094264da60a1d96711015a7951eb2655363309d790ab44ee7b26adff8385e2c210b8268c5ae
de5f82f268554a6fc22ece6feeee2a6264706e71416a0dbe8c39ceacd86054d7cc34dda4fffea4605c037d321290556
10821af85dd9819a4d7e4baa70c51addcda720d33bc9f8bbde6d292c028b2f525a0275ebea968c26f8f0c4b604719ae
3b04e71ed7a8188cd6adf68764374b29c91df3d101a941bf8b7189485ad72
Signature algorithm:   SHA256WithRSASignature
Key size:              2048
Issuer:                C=US, O=IBM, OU=SKLMNode, SKLMCell, Root Certificate, CN=c40bbc1xn3.gpfs.net
Subject:               C=US, O=IBM, OU=SKLMNode, SKLMCell, CN=c40bbc1xn3.gpfs.net

Serial number:          01022a24475466
SHA-256 digest:        077c3b53c5046aa893b760c11cca3a993efbc729479771e03791f9ed4f716879
Signature:              227b5bef89f2e55ef628da6b50db1ab842095a54e1505655e3d95fee753a7f7554868a
a79b294c503dc34562cf69c2a2012879675883896856c0812c4aedd0543d396646a269c02bf4c5ce5acba4409a10e
ffbd47ca38ce492698e2dc8390b9ae3f4a47c23ee3045ff0145218668f35a63edac68201789ed0db6e5c170f5c6db
49769f0b4c9a5f208746e4342294c447793ed087fa0ac762588faf420febeb3fca411e4e725bd46476e1f9f44759a69
6573af5dbbc9553218c7083c80440f2e542bf56cc5cc18156cce05ef6c2e5fea2b886c5c1e262c10af18b13ccf38c3
533ba025b97bbe62f271545b2ab5c1f50c1dca45ce504dfccf257362e9b43
Signature algorithm:   SHA256WithRSASignature
Key size:              2048
Issuer:                C=US, O=IBM, OU=SKLMNode, SKLMCell, Root Certificate, CN=c40bbc1xn3.gpfs.net
Subject:               C=US, O=IBM, OU=SKLMNode, SKLMCell, Root Certificate, CN=c40bbc1xn3.gpfs.net

Do you trust the certificate(s) above? (yes/no) yes
```

Figure 28. Example listing for **mmkeyserv server add**

- d) Issue the **mmkeyserv server show** command to verify that the key server is added. The following listing shows that `keyserver01` is created:

```
# mmkeyserv server show
keyserver01
  Type:                  ISKLM
  Hostname:              keyserver01.gpfs.net
  User ID:               SKLMAdmin
  REST port:             9080
  Label:                 1_keyserver01
  NIST:                  on
  FIPS1402:              off
  Backup Key Servers:
```

```

Distribute: yes
Retrieval Timeout: 120
Retrieval Retry: 3
Retrieval Interval: 10000
REST Certificate Expiration: 2033-05-18 17:01:24 (-0400)
KMIP Certificate Expiration: 2021-05-22 22:24:54 (-0400)

```

3. Issue the **mmkeyserv tenant add** command to add a tenant to the key server. The command creates the tenant on the SKLM server if it does not exist.

A *tenant* is an entity on the SKLM server that can contain encryption keys and certificates. SKLM uses the term *device group* instead of *tenant*.

- a) Issue the following command to add tenant devG1 to key server keyserver01. Enter the SKLM administrator password when prompted:

```
# mmkeyserv tenant add devG1 --server keyserver01
Enter password for the key server keyserver01:
```

- b) Issue the **mmkeyserv tenant show** command to verify that the tenant is added. The following listing shows that tenant devG1 is added to keyserver01:

```
# mmkeyserv tenant show
devG1
  Key Server: keyserver01.gpfs.net
  Registered Client: (none)
```

4. Issue the **mmkeyserv client create** command to create a key client. A key client can request master encryption keys from a tenant after it is registered to the tenant. The command creates a client keystore on the node from which the command is issued and puts into the keystore a set of client credentials and the certificate chain of the SKLM server. The command then copies the keystore to all the nodes in the cluster.

The keystore is stored in the following directory on each node of the cluster:

```
/var/mmfs/ssl/keyServ
```

- a) Issue the following command to create key client c1Client1 for key server keyserver01. Enter the SKLM administrator password and a passphrase for the new keystore when prompted:

```
# mmkeyserv client create c1Client1 --server keyserver01
Enter password for the key server keyserver01:
Create a pass phrase for keystore:
Confirm your pass phrase:
```

Alternatively, issue the following command to create key client c1Client1 for key server keyserver01 using a user-provided, CA-signed certificate. The client certificate file is `client1CertFile.cert`, the client's key file is `client1PrivFile.pem`, and the CA chain file is `CACertChain.pem`. Enter the SKLM administrator password and a passphrase for the new keystore when prompted:

```
# mmkeyserv client create c1Client1 --server keyserver01 -cert client1CertFile.cert
  --priv client1PrivFile.pem --ca-chain CACertChain.pem
Enter password for the key server keyserver01:
Create a pass phrase for keystore:
Confirm your pass phrase:
```

There are three elements to using external certificates:

- A CA-signed certificate file, which certifies the client's identity.
- A private key file that matches the client's certificate.
- The certificate chain of the CA that signed the client certificate.

All these elements must be provided to the **mmkeyserv** command to establish trust in the client's identity and to use it to create a secure connection with the SKLM server. The certificates must be in PEM-encoded x509 format, and the content of the private key file must be PEM-encoded and unencrypted.

The CA certificate chain can be used either as individual files, one file for each CA certificate in the chain, or as a chain file that contains all the CA certificates:

- To create a chain file, concatenate all the CA certificates from the certificate authority into a single file. The file must begin with the CA root certificate, continue with the intermediate CA certificates in the order in which they are used, and end with the CA certificate that signed the client certificate.
- To use the CA certificates as individual files, copy them to a temporary location and rename each file using the format <CACertFilesPrefix>. <n>.cert, where <CACertFilesPrefix> is the full path prefix for the CA certificate files, such as /tmp/CA/certfiles, and <n> is a CA certificate index. The index is 0 for the CA root certificate and n - 1 for the last intermediate CA certificate that signed the client certificate.

In the following example, the chain consists of a CA root certificate file and two intermediate CA certificate files. The full path prefix is /tmp/CA/certfiles:

CA root certificate	/tmp/CA/certfiles.0.cert
First intermediate CA root certificate:	/tmp/CA/certfiles.1.cert
Second intermediate CA root certificate:	/tmp/CA/certfiles.2.cert

Issue the following command to create key client c1Client1 for key server keyserver01:

```
# mmkeyserv client create c1Client1 --server keyserver01 --cert client1CertFile.cert --priv client1PrivFile.pem --ca-cert /tmp/CA/certfiles  
Enter password for the key server keyserver01:  
Create a pass phrase for keystore:  
Confirm your pass phrase:
```

- b) Issue the **mmkeyserv client show** command to verify that the key client is created. The Certificate Type attribute is set to user-provided if the client was created with a CA-signed certificate or to system-generated if the client was created with a self-signed certificate that was generated by IBM Storage Scale. In the following example, the output shows that key client c1Client1 was created for remote key server keyserver01.gpfs.net and that the client certificate is a system-generated, self-signed certificate:

```
# mmkeyserv client show  
c1Client1  
Label: c1Client1  
Key Server: keyserver01.gpfs.net  
Tenants: (none)  
Certificate Expiration: 2023-03-11 00:01:03 (-0500)  
Certificate Type: system-generated
```

In the following example, the output shows that key client c1Client1 was created with a user-provided, CA-signed certificate:

```
# mmkeyserv client show  
c1Client1  
Label: c1Client1  
Key Server: keyserver01.gpfs.net  
Tenants: (none)  
Certificate Expiration: 2023-03-11 00:01:03 (-0500)  
Certificate Type: user-provided
```

5. Issue the **mmkeyserv client register** command to register the key client with the tenant:

You must provide a remote key management (RKM) ID as an input for this command. The RKM ID will become the identifier field of a new RKM stanza that describes the connection between this key client, this tenant, and this key server. For more information about the RKM stanza, see “[The RKM.conf file and the RKM stanza](#)” on page 744.

It is a good practice to use a format like the following one to ensure that the RKM ID is unique:

```
keyServerName_tenantName
```

For example, the RKM ID for the key server and the tenant in these instructions is keyserver01_devG1.

- a) Issue the following command to register key client c1Client1 with tenant devG1 under RKM ID keyserver01_devG1. Enter the requested information when prompted:

```
# mmkeyserv client register c1Client1 --tenant devG1 --rkm-id keyserver01_devG1
Enter password for the key server:
mmkeyserv: [I] Client currently does not have access to the key. Continue the
registration
process ...
mmkeyserv: Successfully accepted client certificate
```

- b) Issue the command **mmkeyserv tenant show** to verify that the key client is known to the tenant.

The following listing shows that tenant devG1 lists c1Client1 as a registered client:

```
mmkeyserv tenant show
devG1
Key Server:      keyserver01.gpfs.net
Registered Client: c1Client1
```

- c) You can also issue the command **mmkeyserv client show** to verify that the tenant is known to the client.

The following listing shows that client c1Client1 is registered with tenant devG1:

```
# mmkeyserv client show
c1Client1
Label:          c1Client1
Key Server:    keyserver01.gpfs.net
Tenants:        devG1
Certificate Expiration: 2023-03-11 00:01:03 (-0500)
```

- d) To see the contents of the RKM stanza, issue the **mmkeyserv rkm show** command.

In the following listing, notice that the RKM ID of the stanza is keyserver01_devG1, the string that was specified in Step 5(a):

```
# mmkeyserv rkm show
keyserver01_devG1 {
type = ISKLM
kmipServerUri = tls://192.0.2.59:5696
keyStore = /var/mmfs/ssl/keyServ/serverKmip.1_keyserver01.c1Client1.1.p12
passphrase = pw4c1Client1
clientCertLabel = c1Client1
tenantName = devG1
}
```

- e) You can also see the RKM stanza by displaying the contents of the RKM.conf file on the node:

```
# cat /var/mmfs/ssl/keyServ/RKM.conf
keyserver01_devG1 {
type = ISKLM
kmipServerUri = tls://192.0.2.59:5696
keyStore = /var/mmfs/ssl/keyServ/serverKmip.1_keyserver01.c1Client1.1.p12
passphrase = pw4c1Client1
clientCertLabel = c1Client1
tenantName = devG1
}
```

6. Issue the **mmkeyserv key create** command to create a master encryption key in the tenant. The following command creates a master encryption key in tenant devG1 of server keyserver01.gpfs.net.

The command displays the UUID of the encryption key (not the key value itself) at line 3 of the listing:

```
# mmkeyserv key create --server keyserver01.gpfs.net --tenant devG1
Enter password for the key server keyserver01.gpfs.net:
KEY-d4e83148-e827-4f54-8e5b-5e1b5cc66de1
```

7. Set up an encryption policy on the node.

- a) Create a file management policy that instructs GPFS to do the encryption tasks that you want.

The following example policy instructs IBM Storage Scale to encrypt all files in the file system with a file encryption key (FEK) and to wrap the FEK with a master encryption key (MEK):

```
RULE 'p1' SET POOL 'system' /* one placement rule is required at all times */
RULE 'Encrypt all files in file system with rule E1'
SET ENCRYPTION 'E1'
WHERE NAME LIKE '%'
RULE 'simpleEncRule' ENCRYPTION 'E1' IS
ALGO 'DEFAULTNISTSP800131A'
KEYS ('KEY-d4e83148-e827-4f54-8e5b-5e1b5cc66de1:keyserver01_devG1')
```

In the last line of the policy, the character string within single quotation marks ('') is the key name. A *key name* is a compound of two parts in the following format:

KeyID:*RkmID*

where:

KeyID

Specifies the UUID of the key that you created in Step 6.

RkmID

Specifies the RKM ID that you specified in Step 5(a).

- b) Issue the **mmchpolicy** command to install the rule.



CAUTION: Installing a new policy with the **mmchpolicy** command removes all the statements in the previous policy. To add statements to an existing policy without deleting the previous contents, collect all policy statements for the file system into one file. Add the new statements to the file and install the contents of the file with the **mmchpolicy** command.

- i) Issue the following command to install the policy rules in file enc.pol for file system c1FileSystem1:

```
# mmchpolicy c1FileSystem1 /tmp/enc.pol
Validated policy 'enc.pol': Parsed 3 policy rules.
Policy 'enc.pol' installed and broadcast to all nodes.
```

- ii) You can list the new encryption policy with the following command:

```
# mm1spolicy c1FileSystem1 -L
```

8. Test the new encryption policy:

- a) Create a file in the file system c1FileSystem1:

```
# echo 'Hello World!' >/c1FileSystem1/hw.enc
```

The policy engine detects the new file, encrypts it, and wraps the file encryption key in a master encryption key.

- b) To verify that the file hw.enc is encrypted, issue the following command to display the encryption attribute of the file.

The output shows that the file is encrypted:

```
# mm1sattr -n gpfs.Encryption /c1Filesystem1/hw.enc
file name:          /c1Filesystem1/hw.enc
gpfs.Encryption:   "EAGC????.????????????? ?????h????????????????? ?u?~?}?????????????
t??1N??
'k???*?3??C???#?)?KEY-ef07b465-cfa5-4476-9f63-544e4b3cc119?NewGlobal11?"
EncPar 'AES:256:XTS:FEK:HMACSHA512'
      type: wrapped FEK WrpPar 'AES:KWRAP' CmbPar 'XORHMACSHA512'
      KEY-d4e83148-e827-4f54-8e5b-5e1b5cc66de1:keyserver01_devG1
```

Part 4: Adding a node to the cluster

- When you add a node to a cluster that is configured by the simplified setup, the cluster automatically detects the new node and copies the encryption configuration to it. For other requirements, see the Requirements section earlier in the topic.

Simplified setup: Valid and invalid configurations

Considerable flexibility and a few restrictions govern the registering of key clients with tenants.

Single cluster, single key server

With a single cluster and a single key server, the following rules apply:

- A single key client can register with more than one tenant.
- However, two or more key clients cannot register with the same tenant.

The following figure illustrates these rules:

- Key client c1Client1 can register with tenants devG1, devG2, and devG3.
- But key client c1Client2 cannot register with devG1 (or devG2 or devG3) because c1Client1 is already registered there.
- Tenant devG4 is added so that key client c1Client2 can register with a tenant.

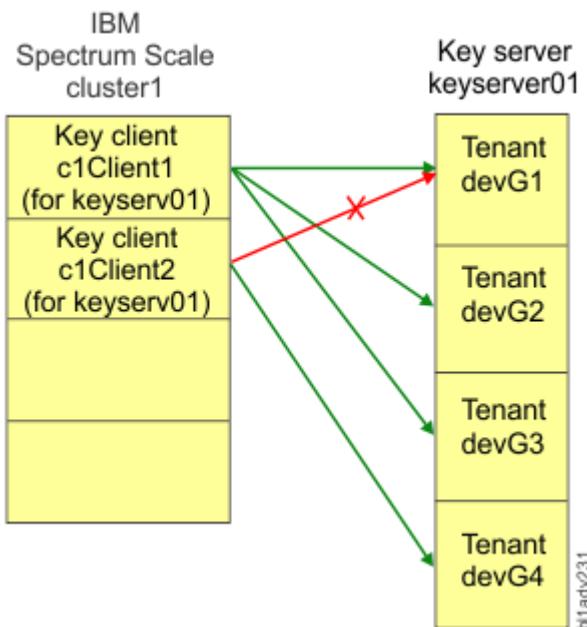


Figure 29. Single cluster, single key server

Multiple clusters, single key server

With multiple clusters and a single key server, more than one key client can register with a tenant if the key clients are in different clusters.

The following figure illustrates these rules:

- With key clients c1Client1 in Cluster1 and c2Client1 in Cluster2:
 - c1Client1 is registered with tenants devG1, devG2, and devG3.
 - c2Client1 can also register with devG1, devG2, and devG3, because it is in a different cluster.
- Similarly, with c1Client2 in Cluster1 and c2Client1 in Cluster2:

- c1Client2 is registered with tenant devG4.
- c2Client1 can also register with devG4, because c2Client1 is in a different cluster.

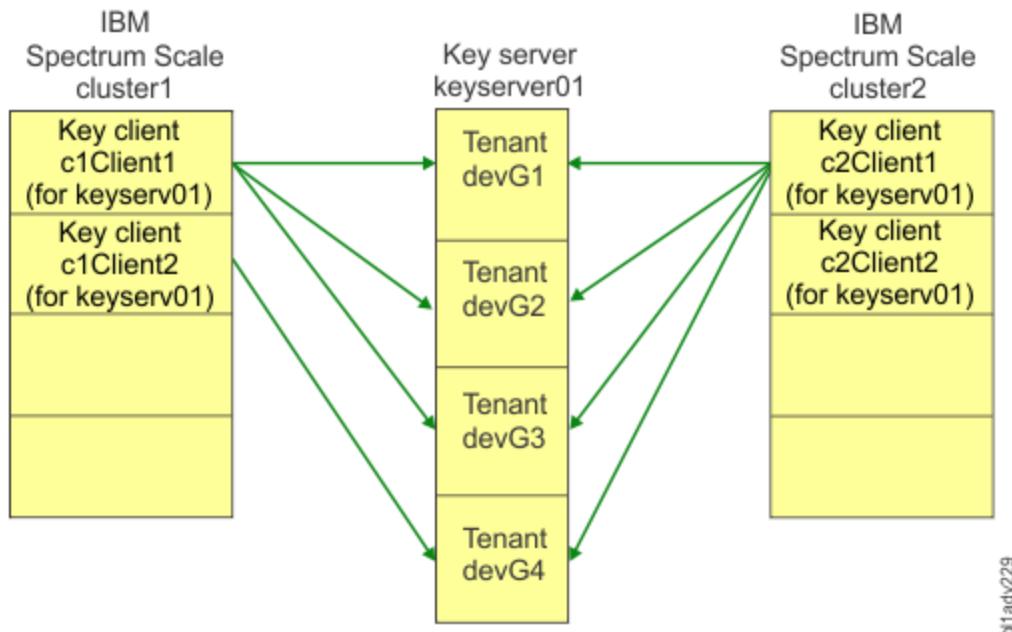


Figure 30. Multiple clusters, single key server

Single cluster, multiple key servers

With a single cluster and multiple key servers, the following rules apply:

- Different key clients in the same cluster can register with different tenants in the same key server.
- But a single key client cannot register with tenants in different key servers.

The following figure illustrates these rules:

- With key clients c1Client1 and c1Client2, both in Cluster1, it is the same situation as in [Figure 29 on page 772](#).
 - c1Client1 is registered with tenants devG1, devG2, and devG3 in keyserver01.
 - c1Client2 can register with tenant devG4 in (but not with devG1, devG2, or devG3).
- With key client c1Client2 in Cluster1:
 - c1Client2 can register with a tenant (devG4 in this example) in.
 - But c1Client2 cannot also register with a tenant (devG3) in keyserver02.
- c1Client3 was created in Cluster1 to register with tenants devG1 and devG2 in keyserver02.

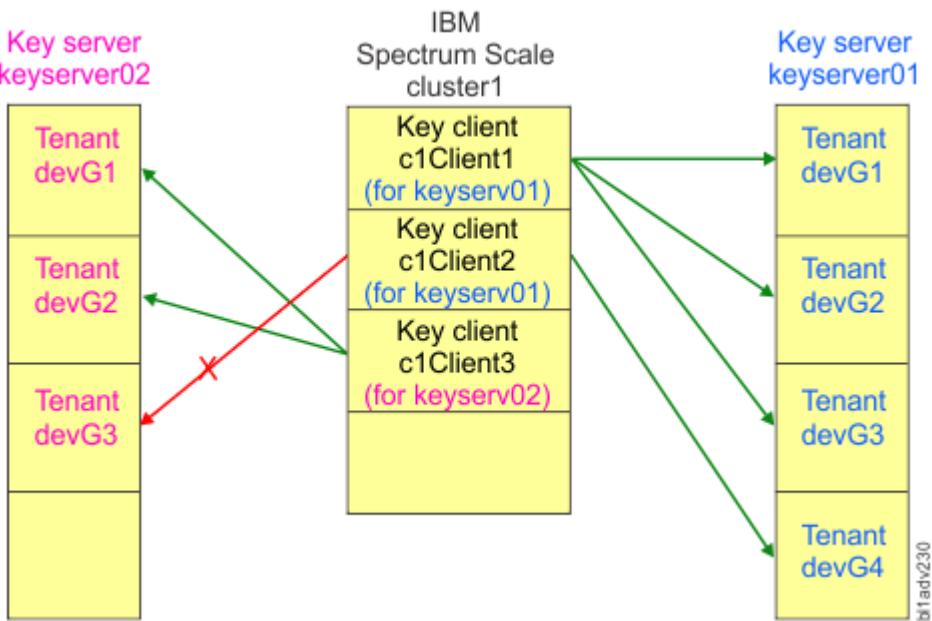


Figure 31. Single cluster, multiple key servers

Simplified setup: Accessing a remote file system

See an example of how to access an encrypted file in a remote cluster.

This topic shows how to configure a cluster so that it can mount an encrypted file system that is in another cluster. In the examples in this topic, the encrypted file system is c1FileSystem1 and its cluster is Cluster1. The cluster that mounts the encrypted file system is Cluster2.

The examples assume that Cluster1 and c1FileSystem1 are the cluster and file system that you configured in the topic “[Simplified setup: Using SKLM with a self-signed certificate](#)” on page 747. You configured Cluster1 for encryption and you created a policy that caused all the files in c1FileSystem1 to be encrypted.

To configure Cluster2 with remote access to an encrypted file in Cluster1, you must configure Cluster2 for encryption in much the same way that Cluster1 was configured. As the following table shows, Cluster2 must add the same key server and tenant as Cluster1. However, Cluster2 must create its own key client and register it with the tenant.

Note: In the third column of the table, items in square brackets are connected or added during this topic. The fourth column shows the step in which each item in the third column is added.

Table 62. Setup of Cluster1 and Cluster2			
Item	Cluster1	Cluster2	Steps
File system	c1FileSystem1	[c1FileSystem1_Remote]	Step 1
Connected to a key server	keyserver01	[keyserver01]	Step 2
Connected to a tenant	c1Tenant1 on keyserver01	[c1Tenant1 on keyserver01]	Step 3
Created a key client	c1Client1	[c2Client1]	Step 4
Registered the key client to the tenant	c1Client1 to c1Tenant1	[c2Client1 to c1Tenant1]	Step 5

Table 62. Setup of Cluster1 and Cluster2 (continued)

Item	Cluster1	Cluster2	Steps
Has access to master encryption keys	c1Client1	[c2Client1]	Step 6
Has access to encrypted file	Local access to hw.enc in c1FileSystem1	[Remote access to hw.enc in c1FileSystem1.]	Step 6

The encrypted file hw.enc is in c1FileSystem1 on Cluster1. To configure Cluster2 to have remote access to file hw.enc, follow these steps:

1. From a node in Cluster2, connect to the remote Cluster1:

- To set up access to the remote cluster and file system, follow the instructions in topic [Chapter 38, “Accessing a remote GPFS file system,” on page 505.](#)
- Run the mmremotefs add command to make the remote file system c1FileSystem1 known to the local cluster, Cluster2:

Note: c1FileSystem1_Remote is the name by which the remote file system c1FileSystem1 is known to Cluster2.

```
# mmremotefs add c1FileSystem1_Remote -f c1FileSystem1 -C Cluster1.gpfs.net -T
/c1FileSystem1_Remote -A no
mmremotefs: Propagating the cluster configuration data to all affected nodes.
This is an asynchronous process.
Tue Mar 29 06:38:07 EDT 2016: mmcommon pushSdr_async: mmsdrfs propagation started.
```

Note: After you have completed Step 1(b) and mounted the remote file system, if you try to access the contents of file hw.enc from Cluster2, the command fails because the local cluster does not have the master encryption key for the file:

```
# cat /c1FileSystem1_Remote/hw.enc
cat: hw.enc: Operation not permitted

mmfs.log:
Tue Mar 29 06:39:27.306 2016: [E]
Key 'KEY-d4e83148-e827-4f54-8e5b-5e1b5cc66de1:keyserver01_devG1'
could not be fetched. The specified RKM ID does not exist;
check the RKM.conf settings.
```

2. From a node in Cluster2, connect to the same SKLM key server, keyserver01, that Cluster1 is connected to:

- Run the mmkeyserv server add to connect to keyserver01:

```
# mmkeyserv server add keyserver01
Enter password for the key server keyserver01:
The security certificate(s) from keyserver01.gpfs.net must be accepted to continue.

View the certificate(s) to determine whether you want to trust the certifying authority.
Do you want to view or trust the certificate(s)? (view/yes/no) view

Serial number:          01022a8adf20f3
SHA-256 digest:        2ca4a48a3038f37d430162be8827d91eb584e98f5b3809047ef4a1c72e15fc4c
Signature:              7f0312e7be18efd72c9d8f37dbb832724859ba4bb5827c230e2161473e0753b367ed49d
993505bd23858541475de8e021e0930725abbd3d25b71edc8fc3de20b7c2db5cd4e865f41c7c410c1d710acf22
2e1c4
5189108e40568ddcbef21094264da60a1d96711015a7951eb2655363309d790ab44ee7b26adf8385e2c210b826
8c5ae
de5f82f268554a6fc22ce6feeee2a6264706e71416a0dbe8c39ceacd86054d7cc34dda4fffea4605c037d3212
90556
10821af85dd9819a4d7e4baa70c51addcda720d33bc9f8bbde6d292c028b2f525a0275ebea968c26f8f0c4b604
719ae
3b04e71ed7a8188cd6adf68764374b29c91df3d101a941bf8b7189485ad72
Signature algorithm:   SHA256WithRSASignature
Key size:              2048
Issuer:                C=US, O=IBM, OU=SKLMNode, SKLMCell, Root Certificate,
```

```

CN=c40bbc1xn3.gpfs.net
Subject: C=US, O=IBM, OU=SKLMNode, SKLMCell, CN=c40bbc1xn3.gpfs.net

Serial number: 01022a24475466
SHA-256 digest: 077c3b53c5046aa893b760c11cca3a993efbc729479771e03791f9ed4f716879
Signature:
227b5bef89f2e55ef628da6b50db1ab842095a54e1505655e3d95fee753a7f7554868a
a79b294c503dc34562cf69c2a20128796758838968565c0812c4aedbb0543d396646a269c02bf4c5ce5acba440
9a10e
ffbd47ca38ce492698e2cdc8390b9ae3f4a47c23ee3045ff0145218668f35a63edac68201789ed0db6e5c170f
5c6db
49769f0b4c9a5f208746e4342294c447793ed087fa0ac762588faf420febeb3fc411e4e725bd46476e1f9f447
59a69
6573af5dbbc9553218c7083c80440f2e542bf56cc5cc18156cce05efd6c2e5fea2b886c5c1e262c10af18b13cc
f38c3
533ba025b97bbe62f271545b2ab5c1f50c1dca45ce504dfcf257362e9b43
Signature algorithm: SHA256WithRSASignature
Key size: 2048
Issuer: C=US, O=IBM, OU=SKLMNode, SKLMCell, Root Certificate,
CN=c40bbc1xn3.gpfs.net
Subject: C=US, O=IBM, OU=SKLMNode, SKLMCell, Root Certificate,
CN=c40bbc1xn3.gpfs.net

Do you trust the certificate(s) above? (yes/no) yes

```

b) Verify that the connection succeeded:

```

# mmkeyserv server show
keyserver01.gpfs.net
  Type: ISKLM
  IPA: 192.168.40.59
  User ID: SKLMAdmin
  REST port: 9080
  Label: 1_keyserver01
  NIST: on
  FIPS1402: off
  Backup Key Servers:
  Distribute: yes
  Retrieval Timeout: 120
  Retrieval Retry: 3
  Retrieval Interval: 10000

```

3. From a node in Cluster2, add the same tenant, c1Tenant1, that Cluster1 added:

a) Add the tenant devG1:

```

# mmkeyserv tenant add devG1 --server keyserver01
Enter password for the key server keyserver01:
mmkeyserv: [I] Tenant devG1 belongs to GPFS family exists on the key server.
Processing continues ...
mmkeyserv: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.

```

b) Verify that the tenant is added:

```

# mmkeyserv tenant show
devG1
  Key Server: keyserver01.gpfs.net
  Registered Client: (none)

```

4. From a node in Cluster2, create a key client:

a) Create the key client c2Client1:

```

# mmkeyserv client create c2Client1 --server keyserver01
Enter password for the key server keyserver01:
Create a pass phrase for keystore:
Confirm your pass phrase:
mmkeyserv: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.

```

b) Verify that the key client is created:

```

# mmkeyserv client show
c2Client1
  Label: c2Client1

```

Key Server:	keyserver01.gpfs.net
Tenants:	(none)

5. From a node in Cluster2, register the key client to the same tenant that Cluster1 is registered to. The RKM ID must be the same as the one that Cluster1 uses, to allow files created with that RKM ID on Cluster1 to be accessed from Cluster2. However, some of the information in the RKM stanza is different:

- a) Register the client in Cluster2 to the same tenant c1Tenant1:

```
# mmkeyserv client register c2Client1 --tenant devG1 --rkm-id keyserver01_devG1
Enter password for the key server :
mmkeyserv: [I] Client currently does not have access to the key.
Continue the registration process ...
mmkeyserv: Successfully accepted client certificate
mmkeyserv: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

- b) Verify that the tenant shows that c2Client1 is registered:

```
# mmkeyserv tenant show
devG1
    Key Server:      keyserver01.gpfs.net
    Registered Client:   c2Client1
```

- c) Verify that c2Client1 shows that it is registered to the c1Tenant:

```
# mmkeyserv client show
c2Client1
    Label:          c2Client1
    Key Server:    keyserver01.gpfs.net
    Tenants:       devG1
```

- d) You can display the contents of the new RKM stanza:

```
# mmkeyserv rkm show
keyserver01_devG1 {
    type = ISKLM
    kmipServerUri = tls://192.168.40.59:5696
    keyStore = /var/mmfss/ssl/keyServ/serverKmip.1_keyserver01.c2Client1.1.p12
    passphrase = c2Client1
    clientCertLabel = c2Client1
    tenantName = devG1
}
```

- e) You can also view the RKM stanza by displaying the contents of the RKM.conf file on the command-line console:

```
# cat /var/mmfss/ssl/keyServ/RKM.conf
keyserver01_devG1 {
    type = ISKLM
    kmipServerUri = tls://192.168.40.59:5696
    keyStore = /var/mmfss/ssl/keyServ/serverKmip.1_keyserver01.c2Client1.1.p12
    passphrase = pwc2Client1
    clientCertLabel = c2Client1
    tenantName = devG1
}
```

6. You can now access the encrypted file hw.enc remotely from Cluster2:

- a) Verify that you can access the contents of the file hw.enc:

```
# cat /c1FileSystem1_Remote/hw.enc
Hello World!
```

- b) Display the encryption attributes of the file:

```
# mmIsattr -n gpfs.Encryption /c1FileSystem1_Remote/hw.enc
file name: /c1FileSystem1_Remote/hw.enc
gpfs.Encryption: "EAGC????t!v???????????? ??????=T????????????? ???O??3????)??r??nV?K?
OA?;?????
??x,?:w?d????KEY-d4e83148-e827-4f54-8e5b-5e1b5cc66de1?keyserver01_devG1?"
EncPar 'AES:256:XTS:FEK:MACSHA512'
type: wrapped FEK
```

```

WrPPar 'AES:KWRAP'
CmbPar 'XORHMACSHA512'
KEY-d4e83148-e827-4f54-8e5b-5e1b5cc66de1:keyserver01_devG1

```

You can now access encrypted files on c1FileSystem1_Remote from Cluster2.

Accessing an encrypted remote file system using keys from Vault KMIP Secrets Engine

See an example of how to access an encrypted file in a remote cluster.

This topic shows how to configure a cluster so that it can mount an encrypted file system that is in another cluster. In the examples in this topic, the encrypted file system is fs1 and its cluster is Cluster1. The cluster that mounts the encrypted file system is Cluster2.

The examples assume that Cluster1 and fs1 are the cluster and file system that you configured in the topic [“Setup using HashiCorp Vault KMIP Secrets Engine” on page 757](#). You configured Cluster1 for encryption and you created a policy that caused all the files in fs1 to be encrypted.

To configure Cluster2 with remote access to an encrypted file in Cluster1, you must configure Cluster2 for encryption in much the same way that Cluster1 was configured. As the following table shows, Cluster2 must add the same key server, create role in the same scope name and register the role using the same RKM stanza name.

Note: In the third column of the table, items in square brackets are connected or added during this topic. The fourth column shows the step in which each item in the third column is added.

Table 63. Setup of Cluster1 and Cluster2			
Item	Cluster1	Cluster2	Steps
File systems	fs1	[fs1_remote]	Step 1
Connected to key server	tru-4pub.fyre.ibm.com	[tru-4pub.fyre.ibm.com]	Step 2
Scope name	spectrumscale (default)	[spectrumscale] (default)	Step 3
Created role	gpfsAdmin	[gpfsAdminRemote]	Step 4
Registered the role to RKM stanza	gpfsAdmin to gpfsRKMstanza	[gpfsAdminRemote to gpfsRKMstanza]	Step 4
Has access to master encryption keys	gpfsAdmin role in spectrumscale scope	[gpfsAdmin role in spectrumscale]	Step 4
Has access to encrypted file	Local access to testFile file in fs1 file system	[Remote access to file testFile in file system fs1]	Step 5

The encrypted file testFile is in fs1 on Cluster1. To configure Cluster2 to have remote access to file testFile, follow the below steps:

1. From a node in Cluster2, connect to the remote Cluster1:
 - a) To set up access to the remote cluster and file system, follow the instructions in topic [Chapter 38, “Accessing a remote GPFS file system,” on page 505](#).
 - b) Run the mmremotefs add command to make the remote file system fs1 known to the local cluster, Cluster2:

Note: fs1_Remote is the name by which the remote file system fs1 is known to Cluster2.

```

# mmremotefs add fs1_remote -f fs1 -C Cluster1.gpfs.net -T /fs1_remote -A no
mmremotefs: mmsdrfs propagation completed.

```

Note: After you have completed Step 1(b) and mounted the remote file system, if you try to access the contents of file `testFile` from Cluster2, the command fails because the local cluster does not have the master encryption key for the file:

```
# mmount fs1_remote
Wed Jul 13 00:31:32 EDT 2022: mmount: Mounting file systems ...

# cat /fs1_remote/testFile
cat: /fs1_remote/testFile: Operation not permitted

mmfs.log:
2022-07-13_00:31:53.456-0400: [E] Unable to open encrypted file: inode 65792, fileset 0,
file system fs1.
2022-07-13_00:31:53.456-0400: [E] Key 'leCTiYY6fUPCgQsk5SHFBtTgADJHgax:gpfsRKMstanza'
could not be fetched. The specified RKM ID does not exist; check the RKM.conf settings.
```

2. From a node in Cluster2, connect to the Vault key server, `tru-4pub.fyre.ibm.com`, that Cluster1 is connected to.

- a) Run the `mmkeyserv server add` to connect to `tru-4pub.fyre.ibm.com`:

```
# mmkeyserv server add tru-4pub.fyre.ibm.com --auth-token /var/mmfs/ssl/keyServ/tmp/
vaultTempToken
```

- b) Verify that the connection is succeeded:

```
# mmkeyserv server show
tru-4pub.fyre.ibm.com
  Type: KMIP
  IPA: 9.46.79.137
  User ID: N/A
  REST port: 8200
  Label: 1_tru-4pub
  NIST: on
  FIPS1402: off
  Backup Key Servers:
  Distribute: yes
  Retrieval Timeout: 60
  Retrieval Retry: 3
  Retrieval Interval: 10000
  REST Certificate Expiration: N/A
  KMIP Certificate Expiration: N/A
```

3. From a node in Cluster2, create a role in the same scope, `spectrumscale` (the default), that Cluster1 created.

- a) Create the `gpfsAdminRemote` role:

```
# mmkeyserv role create gpfsAdminRemote --server tru-4pub.fyre.ibm.com --auth-token
tempToken
Create a pass phrase for keystore:
Confirm your pass phrase:
mmkeyserv: mmsdrfs propagation completed.
```

- b) Verify that the role is created:

```
# mmkeyserv role show
gpfsAdminRemote
  Key Server: tru-4pub.fyre.ibm.com
  Scope: spectrumscale
  Role Label: 1_gpfsAdminRemote
  RKM Id:
  CA Chain Expiration: 2032-05-11 12:40:00 (-0400)
  Certificate Expiration: 2025-07-12 00:59:47 (-0400)
  Certificate Serial Number: 302209357934438088530728828729422923497833539651
  Certificate Type: system-generated
```

4. From a node in Cluster2, register the `gpfsAdminRemote` role.

The RKM ID must be the same as the one that Cluster1 uses, to allow files created with that RKM ID on Cluster1 to be accessed from Cluster2. However, some of the information in the RKM stanza is different:

- a) Register the `gpfsAdminRemote` role in Cluster2 using the same RKM stanza:

```
# mmkeyserv role register gpfsAdminRemote --rkm-id gpfsRKMstanza
mmkeyserv: mmsdrfs propagation completed.
```

- b) Verify that the role shows that it is registered:

```
# mmkeyserv role show
gpfsAdminRemote
  Key Server: tru-4pub.fyre.ibm.com
  Scope: spectrumscale
  Role Label: 1_gpfsAdminRemote
  RKM Id: gpfsRKMstanza
  CA Chain Expiration: 2032-05-11 12:40:00 (-0400)
  Certificate Expiration: 2025-07-12 00:59:47 (-0400)
  Certificate Serial Number: 302209357934438088530728828729422923497833539651
  Certificate Type: system-generated
```

- c) You can display the contents of the new RKM stanza:

```
# mmkeyserv rkm show
gpfsRKMstanza {
  type = KMIP
  kmipServerUri = tls://9.46.79.137:5696
  keyStore = /var/mmfs/ssl/keyServ/roleCred.1_gpfsAdminRemote.1.p12
  passphrase = pass!@#ForDemo
  clientCertLabel = 1_gpfsAdminRemote
}
```

- d) You can also view the RKM stanza by displaying the contents of the RKM.conf file on the command-line console:

```
# cat /var/mmfs/ssl/keyServ/RKM.conf
gpfsRKMstanza {
  type = KMIP
  kmipServerUri = tls://9.46.79.137:5696
  keyStore = /var/mmfs/ssl/keyServ/roleCred.1_gpfsAdminRemote.1.p12
  passphrase = pass!@#ForDemo
  clientCertLabel = 1_gpfsAdminRemote
}
```

5. You can now access the encrypted the testFile file remotely from Cluster2:

- a) Verify that you can access the contents of the testFile file:

```
# cat /onFilefs1_remote/testFile
Hello World!
```

- b) Display the encryption attributes of the file:

```
# mmIsattr -n gpfs.Encryption /onFilefs1_remote/testFile
file name: /onFilefs1_remote/testFile
gpfs.Encryption: "EAGC????*Z????????????? ??????C>yA????????????? ?2+]???ie???aq?q?
Am?W0?Q??\?8?@??t??1UN?!?lectiYYs6fUPCgQsk5SHFBtTgADJHgax?gpfsRKMstanza?"
EncPar 'AES:256:XTS:FEK:HMACSHA512'
  type: wrapped FEK WrpPar 'AES:KWRAP' CmbPar 'XORHMACSHA512'
    leCTiYYs6fUPCgQsk5SHFBtTgADJHgax:gpfsRKMstanza
```

You can now access encrypted files on fs1_remote from Cluster2.

Simplified setup: Doing other tasks

Learn how to do other tasks after you complete the simplified setup.

For the first three tasks in this topic, you need the password for your SKLM key server.

[“Adding a node to the cluster” on page 781](#)

[“Creating encryption keys” on page 781](#)

[“Adding a tenant to GKLM” on page 781](#)

[“Create keys on Vault” on page 782](#)

[“Managing another key server” on page 783](#)

[“Adding backup key servers” on page 786](#)

[“Adding a role or scope to Vault” on page 787](#)

Adding a node to the cluster

- When you add a node to a cluster that is configured for encryption by the simplified setup, the cluster automatically detects the node and copies the encryption configuration to it. To encrypt files, the node must have direct network access to the Remote Key Management (RKM) server. For more information, see the section “Requirements” in the topic [“Preparation for encryption” on page 742](#).

Creating encryption keys

This task shows how to create encryption keys in a tenant:

- The following command creates five encryption keys in tenant devG1 on key server keyserver01 and displays the UUIDs of the keys on the console:

```
# mmkeyserv key create --server keyserver01.gpfs.net --tenant devG1 --count 5
Enter password for the key server keyserver01.gpfs.net:
KEY-492911c8-e3d4-4670-9868-617243d4ca57
KEY-5f24d71f-daf3-4df8-90e4-5f6475370f70
KEY-a487b01d-f092-4895-b537-139edeb57239
KEY-b449b3a2-73c5-499f-b575-fc7ba95541a8
KEY-fd3dbeef9-0e6c-4662-9410-bfe3b73272b9
```

- The following command shows the UUIDs of the encryption keys on tenant devG1 in keyserver01:

```
# mmkeyserv key show --server keyserver01.gpfs.net --tenant devG1
Enter password for the key server keyserver01.gpfs.net:
KEY-492911c8-e3d4-4670-9868-617243d4ca57
KEY-5f24d71f-daf3-4df8-90e4-5f6475370f70
KEY-a487b01d-f092-4895-b537-139edeb57239
KEY-b449b3a2-73c5-499f-b575-fc7ba95541a8
KEY-d4e83148-e827-4f54-8e5b-5e1b5cc66de1
KEY-fd3dbeef9-0e6c-4662-9410-bfe3b73272b9
```

The command displays the UUIDs of the previously existing key and the five new keys.

Adding a tenant to GKLM

A tenant is a container that resides on a key server and contains encryption keys. Before a key client can request master encryption keys from a key server, you must add a tenant to the key server, create a key client, and register the key client with the tenant. For more information, see [“Simplified setup: Using SKLM with a self-signed certificate” on page 747](#).

In some situations, you might need to access more than one tenant on the same key server. For example, if you have several key clients that you want to use with the same key server, each key client must register with a different tenant. For more information, see [“Simplified setup: Valid and invalid configurations” on page 772](#).

This task shows how to add a tenant, register an existing key client with the tenant, and create encryption keys in the tenant.

- Add the tenant:

- Add a tenant devG2 on keyserver01:

```
# mmkeyserv tenant add devG2 --server keyserver01
Enter password for the key server keyserver01:
```

- Verify that the tenant is added. The following command displays all the existing tenants:

```
# mmkeyserv tenant show
devG1
    Key Server:      keyserver01.gpfs.net
    Registered Client: c1Client1

devG2
    Key Server:      keyserver01.gpfs.net
    Registered Client: (none)
```

The tenants are devG1 and devG2.

2. Register the existing key client with the tenant:

a) Register client c1Client1 with tenant devG2:

```
# mmkeyserv client register c1Client1 --tenant devG2 --rkm-id keyserver01_devG2
Enter password for the key server :
mmkeyserv: [I] Client currently does not have access to the key.
Continue the registration process...
mmkeyserv: Successfully accepted client certificate
```

b) Verify that the key client is registered to the tenant:

```
# mmkeyserv client show
c1Client1
    Label:          c1Client1
    Key Server:    keyserver01.gpfs.net
    Tenants:       devG1,devG2
```

The command output shows that c1Client1 is registered to both devG1 and the new devG2.

c) Verify the configuration of the RKM stanza. The following command displays all the RKM stanzas:

```
# mmkeyserv rkm show
keyserver01_devG1 {
    type = ISKLM
    kmipServerUri = tls://192.168.40.59:5696
    keyStore = /var/mmfs/ssl/keyServ/serverKmip.1_keyserver01.c1Client1.1.p12
    passphrase = pw_c1Client1
    clientCertLabel = label_c1Client1
    tenantName = devG1
}
keyserver01_devG2 {
    type = ISKLM
    kmipServerUri = tls://192.168.40.59:5696
    keyStore = /var/mmfs/ssl/keyServ/serverKmip.1_keyserver01.c1Client1.1.p12
    passphrase = pw_c1Client1
    clientCertLabel = label_c1Client1
    tenantName = devG2
}
```

The command shows the following relationships:

- Client c1Client1 is registered with tenant devG1 on keyserver01.
- Client c1Client1 is also registered with tenant devG2 on keyserver01.

3. Create keys in the tenant.

The following command creates three keys in tenant devG2:

```
# mmkeyserv key create --server keyserver01 --tenant devG2 --count 3
Enter password for the key server keyserver01:
KEY-43cf5e69-1640-4056-b114-bdbcf2914189
KEY-4c7540cd-0346-4733-90eb-8df4c0f16008
KEY-c86a523b-e04f-4536-86a6-c6f83f845265
```

Create keys on Vault

This task shows how to create encryption keys on Vault RKM server.

1. The following command uses role gpfsAdmin to create four additional encryption keys in the spectrumscale scope on Vault key server tru-4pub and displays the UUIDs of the keys on the console:

```
# mmkeyserv key create --role gpfsAdmin --count 4
mdA0DrDVZdERX4jaN926kzUSNfPDSk3i
n1nBBOfRQz711zhuuxACZwloRGUcURB
qMCnHy10ILWE4y2BWMKJkfib6t3h5GkJ
t5etBuZuWjnBF42dfpxdWvfuYHq1gq
```

2. The following command shows the UUIDs of the encryption keys:

```
# mmkeyserv key show --role gpfsAdmin
leCTiYY$6fUPCgQsk5SHFBtTgADJHgax
mdA0DrDVZdERX4jaN926kzUSNfPDSk3i
nlnBBofRQz71IzfnuuxACZwlORGUcURB
qMCnHy10ILWE4y2BWMKJkfrb6t3h5GkJ
t5etBuZuWjnBF42dfpxdWvfuYHQq1gqd
```

Managing another key server

This task shows how to add a key server, add a tenant, create a new key client, and register the key client with the tenant. The steps are the same as the ones that you follow in the simplified setup:

Table 64. Managing another key server	
Item	Step
Install and configure SKLM.	Step 1
Add a key server	Step 2
Add a tenant to the key server	Step 3
Create a key client	Step 4
Register the key client with the tenant	Step 5

1. Install and configure IBM Security Key Lifecycle Manager (SKLM).

For more information, see the topic “[Simplified setup: Using SKLM with a self-signed certificate](#)” on [page 747](#).

2. Add the key server, keyserver11. If backup key servers are available, you can add them now. You can have up to five backup key servers.
 - a) Add keyserver11 and backup key servers keyserver12 and keyserver13. Enter the requested information when prompted:

```
# mmkeyserv server add keyserver11 --backup keyserver12,keyserver13
Enter password for the key server keyserver11:
The security certificate(s) from keyserver11.gpfs.net must be accepted to continue.
View the certificate(s) to determine whether you want to trust the certifying authority.
Do you want to view or trust the certificate(s)? (view/yes/no) view

Serial number:          0361e7075056
SHA-256 digest:         2a7ab79d52cca7d2cae6e88077ee48b405a9e87d03d47023fdf1d4e185f18f75
Signature:               55a4350778446ac1f74fe25016bc9efd86893b8c5e9a4c3ebc4662d7cafce8697bfb98
f8ce62ab976fb10270a006074bd36a3c0321bb99417dcfd6d9d18c06ca380f1a89aacf3d0b5d84a7fdde5d4c1b9
377a0
e725d65dee819f489a9c51c2017ac6633304a3973c7e13ddc611aae6d2ba35c8571b6ca1388dbb1b91a51b00f0
9fe37
2846dbe0139e4f942ed317809c0b7d0cd651a3273b4df041719f99847923e5ec58517fd778d46ea44647149c5d
52287
ee9705aa292c1d2942b27dd7f07d6bae2b1f29a4a818655c582ef0ce9102e70a7df68ee0c0732a66b2960959f3
8f964
0c599a3203ff6fcacf13f40e9922fa439d016937a00d0f5a7f571d174f277
Signature algorithm:    SHA256WithRSASignature
Key size:                2048
Issuer:                 C=US, O=IBM, OU=SKLMNode, SKLMCell, Root Certificate,
CN=vmip131.gpfs.net
Subject:                C=US, O=IBM, OU=SKLMNode, SKLMCell, CN=vmip131.gpfs.net

Serial number:          03615d201517
SHA-256 digest:         4acb77202f885f4c6b4c858f701394f18150fd683a0d155885399bbb5b8cc0b1
Signature:               15e2011efd402b4834c677c9bcdca9914f457a9573bf1568c4d309cd1a9b873b857566c
f9653a736e34b63f8e600e1bee2450c838bbf49c6291548f0bb4ee82d8243ba60dcfbcc42f25f965fa36483441
dfe7e
b2089361dbee77e333d2711ee8364f9d5005cf382a42fa90dec8f0e279b5cecb6d5ef3da2d75cdc1e70d7f4545
afc13
547135c4978b717c6572b3d8c569cd44f15c0b084fe92a9e2878bcf34518882c1461e832e014d56d981ad40ef2
c6760
71f49571a91e036c84ab58b3d22d0d971990624751ea6d74a420cfbf2e00d718e263184c97091404d295adb564
67237
```

```

09decacebd7dbfa1927a8143bdf6d6640b72ec7c588b00cf0521c67f6efe9
Signature algorithm: SHA256WithRSASignature
Key size: 2048
Issuer: C=US, O=IBM, OU=SKLMNode, SKLMCell, Root Certificate,
CN=vmip131.gpfs.net
Subject: C=US, O=IBM, OU=SKLMNode, SKLMCell, Root Certificate,
CN=vmip131.gpfs.net

Do you trust the certificate(s) above? (yes/no) yes

```

- b) Verify that the key server is added. The following command displays information about all the existing key servers:

```

# mmkeyserv server show
keyserver01.gpfs.net
    Type: ISKLM
    IPA: 192.168.40.59
    User ID: SKLMAadmin
    REST port: 9080
    Label: 1_keyserver01
    NIST: on
    FIPS1402: off
    Backup Key Servers:
    Distribute: yes
    Retrieval Timeout: 120
    Retrieval Retry: 3
    Retrieval Interval: 10000

keyserver11.gpfs.net
    Type: ISKLM
    IPA: 192.168.9.131
    User ID: SKLMAadmin
    REST port: 9080
    Label: 2_keyserver11
    NIST: on
    FIPS1402: off
    Backup Key Servers: keyserver12.gpfs.net,keyserver13.gpfs.net
    Distribute: yes
    Retrieval Timeout: 120
    Retrieval Retry: 3
    Retrieval Interval: 10000

```

The command shows two key servers, keyserver01 and the keyserver11.

3. Add a tenant to the key server.

The name of the tenant must be unique within the same key server, but it can be the same as the name of a tenant in another key server:

- a) Add the tenant devG1 to keyserver11:

```

mmkeyserv tenant add devG1 --server keyserver11
Enter password for the key server keyserver11:

```

- b) Verify that the tenant is added:

```

mmkeyserv tenant show
devG1
    Key Server: keyserver01.gpfs.net
    Registered Client: c1Client1

devG2
    Key Server: keyserver01.gpfs.net
    Registered Client: c1Client1

devG1
    Key Server: keyserver11.gpfs.net
    Registered Client: (none)

```

The command shows the following tenants:

- Tenant devG1 on keyserver01.
- Tenant devG2 on keyserver01.
- Tenant devG1 on keyserver11.

4. Create a key client:

Note: A key client name must be 1-16 characters in length and must be unique within an IBM Storage Scale cluster.

a) Create c1Client11 on keyserver11.

```
# mmkeyserv client create c1Client11 --server keyserver11
Enter password for the key server keyserver11:
Create a pass phrase for keystore:
Confirm your pass phrase:
```

b) Verify that the client is created. The command shows all the existing key clients:

```
# mmkeyserv client show
c1Client1
  Label:          c1Client1
  Key Server:    keyserver01.gpfs.net
  Tenants:       devG1,devG2

c1Client11
  Label:          c1Client11
  Key Server:    keyserver11.gpfs.net
  Tenants:       (none)
```

The key clients are c1Client1 and c1Client11.

c) You can also display all the clients of keyserver11:

```
# mmkeyserv client show --server keyserver11
c1Client11
  Label:          c1Client11
  Key Server:    keyserver11.gpfs.net
  Tenants:       (none)
```

5. Register the key client with the tenant:

a) Verify that tenant devG1 on keyserver11 has no registered clients:

```
# mmkeyserv tenant show --server keyserver11
devG1
  Key Server:    keyserver11.gpfs.net
  Registered Client: (none)
```

b) Register the key client c1Client11 with the devG1 on keyserver11:

```
# mmkeyserv client register c1Client11 --tenant devG1 --rkm-id keyserver11_devG1
Enter password for the key server of client c1Client11:
mmkeyserv: [I] Client currently does not have access to the key.
Continue the registration process ...
mmkeyserv: Successfully accepted client certificate
```

c) Verify that the tenant shows that the client c1Client11 is registered with it:

```
# mmkeyserv tenant show --server keyserver11
devG1
  Key Server:    keyserver11.gpfs.net
  Registered Client: c1Client11
```

d) You can also verify that the client shows that it is registered with tenant devG1:

```
# mmkeyserv client show --server keyserver11
c1Client11
  Label:          c1Client11
  Key Server:    keyserver11.gpfs.net
  Tenants:       devG1
```

e) Display the RKM stanzas for the cluster. They show the following relationships:

- With keyserver01, c1Client1 is registered with devG1 and devG2.
- With keyserver11, c1Client11 is registered with devG1.

```
# mmkeyserv rkm show
keyserver01_devG1 {
  type = ISKLM
```

```

kmipServerUri = tls://192.168.40.59:5696
keyStore = /var/mmfss/ssl/keyServ/serverKmip.1_keyserver01.c1Client1.1.p12
passphrase = pw4c1Client1
clientCertLabel = c1Client1
tenantName = devG1
}
keyserver01_devG2 {
    type = ISKLM
    kmipServerUri = tls://192.168.40.59:5696
    keyStore = /var/mmfss/ssl/keyServ/serverKmip.1_keyserver01.c1Client1.1.p12
    passphrase = pw4c1Client1
    clientCertLabel = c1Client1
    tenantName = devG2
}
keyserver11_devG1 {
    type = ISKLM
    kmipServerUri = tls://keyserver12.gpfs.net:5696
    kmipServerUri2 = tls://keyserver13.gpfs.net:5696
    kmipServerUri3 = tls://192.168.9.131:5696
    keyStore = /var/mmfss/ssl/keyServ/serverKmip.2_keyserver11.c1Client11.1.p12
    passphrase = pw4c1Client11
    clientCertLabel = c1Client11
    tenantName = devG1
}

```

- f) Create encryption keys. The following command creates two keys in tenant devG1 on keyserver11.

```
# mmkeyserv key create --server keyserver11 --tenant devG1 --count 2
Enter password for the key server keyserver11:
KEY-86f601ba-0643-4f94-92b2-12c8765512cc
KEY-cdcf058f-ae30-41e8-b6f7-754e23322428
```

Adding backup key servers

If multiple key servers exist, you can add them to an RKM stanza to provide backup capability in case the main key server becomes unavailable. You can add up to five backup key servers.

Important: IBM Storage Scale does not manage backup key servers. You must configure them and maintain them.

Note: For information about using backup key servers, see the subtopic "Adding backup RKM servers in a high-availability configuration" in ["Preparation for encryption" on page 742](#).

This task shows how to add backup key servers to the RKM stanza of one of your key clients. You can add backup key servers when you create a key server, as shown in Step 2 of the previous subtopic. Or you can add them later, as in this subtopic.

In this task the primary key server is keyserver11. The backup key servers for the RKM stanza are keyserver12 and keyserver13. You want to add three more backup key servers to the list: keyserver14, keyserver15, and keyserver16.

Follow these steps:

- Add the three backup key servers. You must specify the entire list of key servers, including ones that are already in the list. The following command is on one line. In the list of servers, do not put spaces on either side of the commas (,):

```
# mmkeyserv rkm change keyserver11_devG1 --backup
keyserver12,keyserver13,keyserver14,keyserver15,keyserver16
```



Attention:

- You can change the order in which the client tries backup key servers, by running the same command with the key servers in a different order.
- You can delete backup key servers by specifying a list that contains the backup key servers that you want to keep and omits the ones that you want to delete.

- To verify, issue the **mmkeyserv rkm show** command to display the RKM stanzas:

```
# mmkeyserv rkm show
keyserver01_devG1 {
```

```

type = ISKLM
kmipServerUri = tls://192.168.40.59:5696
keyStore = /var/mmfs/ssl/keyServ/serverKmip.1_keyserver01.c1Client1.1.p12
passphrase = pw4c1Client1
clientCertLabel = c1Client1
tenantName = devG1
}

keyserver01_devG2 {
    type = ISKLM
    kmipServerUri = tls://192.168.40.59:5696
    keyStore = /var/mmfs/ssl/keyServ/serverKmip.1_keyserver01.c1Client1.1.p12
    passphrase = pw4c1Client1
    clientCertLabel = c1Client1
    tenantName = devG2
}

keyserver11_devG1 {
    type = ISKLM
    kmipServerUri = tls://keyserver11.gpfs.net:5696
    kmipServerUri12 = tls://keyserver12.gpfs.net:5696
    kmipServerUri13 = tls://keyserver13.gpfs.net:5696
    kmipServerUri14 = tls://keyserver14.gpfs.net:5696
    kmipServerUri15 = tls://keyserver15.gpfs.net:5696
    kmipServerUri16 = tls://keyserver16.gpfs.net:5696
    keyStore = /var/mmfs/ssl/keyServ/serverKmip.2_keyserver11.c1Client11.1.p12
    passphrase = pw4c1Client11
    clientCertLabel = c1Client11
    tenantName = devG1
}

```

The command output shows the following relationships:

- The configuration of c1Client1, devG1, and keyserver01 has zero backup servers.
- The configuration of c1Client1, devG2, and keyserver01 has zero backup servers.
- The configuration of c1Client11, devG1, and keyserver11 has five backup servers.

Adding a role or scope to Vault

Within an IBM Storage Scale cluster, only one role can be created per scope. The **mmkeyserv role create** command creates a scope and a role at the same time.

Follow these steps:

1. Create the role.
 - a) Create a scope spectrumscale2 and a role gpfsAdmin2:

```
# mmkeyserv role create gpfsAdmin2 --scope spectrumscale2 --server tru-4pub --auth-token
tempToken
Create a pass phrase for keystore:
Confirm your pass phrase:
mmkeyserv: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

- b) To verify the scope and role are created, issue the following command:

```
# mmkeyserv role show gpfsAdmin2
gpfsAdmin2
    Key Server:          tru-4pub.fyre.ibm.com
    Scope:               spectrumscale2
    Role Label:          4_gpfsAdmin2
    RKM Id:
    CA Chain Expiration: 2032-05-11 12:40:00 (-0400)
    Certificate Expiration: 2025-07-12 15:17:04 (-0400)
    Certificate Serial Number: 301049472216982094185931943435573692776800291659
    Certificate Type:     system-generated
```

2. Register the new role.

- a) Register role gpfsAdmin2:

```
# mmkeyserv role register gpfsAdmin2 --rkm-id gpfsRKMstanza2
mmkeyserv: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

- b) Verify that the role is registered:

```
# mmkeyserv role show gpfsAdmin2
gpfsAdmin2
  Key Server: tru-4pub.fyre.ibm.com
  Scope: spectrumscale2
  Role Label: 4_gpfsAdmin2
  RKM Id: gpfsRKMstanza2
  CA Chain Expiration: 2032-05-11 12:40:00 (-0400)
  Certificate Expiration: 2025-07-12 15:17:04 (-0400)
  Certificate Serial Number: 301049472216982094185931943435573692776800291659
  Certificate Type: system-generated
```

- c) Verify the configuration of the RKM stanza. The following command displays all the RKM stanzas:

```
# mmkeyserv rkm show
gpfsRKMstanza {
  type = KMIP
  kmipServerUri = tls://9.46.79.137:5696
  keyStore = /var/mmfs/ssl/keyServ/roleCred.1_gpfsAdmin.1.p12
  passphrase = pass!@#ForDemo
  clientCertLabel = 1_gpfsAdmin
}
gpfsRKMstanza2 {
  type = KMIP
  kmipServerUri = tls://9.46.79.137:5696
  keyStore = /var/mmfs/ssl/keyServ/roleCred.4_gpfsAdmin2.1.p12
  passphrase = pass!@#4admin2
  clientCertLabel = 4_gpfsAdmin2
  connectionTimeout = 15
  connectionAttempts = 4
  retrySleep = 5000
}
```

3. Create a key from new role.

```
# mmkeyserv key create --role gpfsAdmin2
JKua674cMKT1oNdrP7PZQ01wBBc31iVV
```

Regular setup: Using SKLM with a self-signed certificate

Learn to use the regular setup method to configure the key client node with the IBM Security Key Lifecycle Manager (SKLM) key server when the server is running with a self-signed certificate rather than with a certificate chain from a certificate authority (CA).



Attention: The simplified setup method, which can be used only when the Remote Key Management (RKM) server is SKLM, is much easier to use and more powerful than the regular setup method with SKLM. In the simplified setup method, the **mmkeyserv** command automatically performs many of the steps that must be done manually in the regular setup method.

The regular setup with SKLM requires IBM Storage Scale Advanced Edition, IBM Storage Scale Data Management Edition, or IBM Storage Scale Developer Edition or IBM Storage Scale Erasure Code Edition 4.1 or later and a supported version of SKLM. For information about supported SKLM versions, see “[Preparation for encryption](#)” on page 742.

Note: IBM Storage Scale supports IBM Security Guardium Key Lifecycle Manager (GKLM) 4.1.0.1 (IF01), 4.1.1, or later. The older versions of GKLM are referred to as IBM Security Lifecycle Manager or SKLM in the documentation. The configuration information is the same for both GKLM and SKLM.

This topic describes the regular method for setting up encryption with SKLM as the RKM server and with a self-signed certificate on the KMIP port of the SKLM server. If your deployment scenario uses a certificate chain from a CA, see one of the following topics:

[“Simplified setup: Using SKLM with a certificate chain” on page 761](#)

[“Regular setup: Using SKLM with a certificate chain” on page 797.](#)

Note: If you are using SKLM 2.7 or later, see the topic “Configuring encryption with SKLM 2.7 or later” on page 810.

Requirements:

The following requirements must be met on every IBM Storage Scale node that participates in encryption:

- The node must have direct network access to the system where the key server is installed.
- The security-sensitive files that are created during the configuration process must have the following characteristics:
 - They must be regular files that are owned by the root user.
 - The group ownership must be changed to root group.
 - They must be readable and writable only by the user (mode '0600'). The following examples apply to the regular setup with SKLM and with Thales Vormetric Data Security Manager (DSM) setup:

```
-rw----- 1 root root 2446 Mar 20 12:15 /var/mmfs/etc/RKM.conf  
drw----- 2 root root 4096 Mar 20 13:47 /var/mmfs/etc/RKMcerts  
-rw----- 1 root root 3988 Mar 20 13:47 /var/mmfs/etc/RKMcerts/keystore_name.p12
```

These security-sensitive files include the following files:

- The RKM.conf file. For more information about this file, see “[The RKM.conf file and the RKM stanza](#)” on page 744.
- The files in the client keystore directory, which include the keystore file, the public and private key files for the client, and possibly other files. For more information about these files, see “[The client keystore directory and its files](#)” on page 746.



CAUTION:

- Take appropriate precautions to ensure that the security-sensitive files are not lost or corrupted. IBM Storage Scale does not manage or replicate the files.
- Ensure that the passphrase for the client certificate file is not leaked through other means, such as the shell history.
- Client keystore files must be record-locked when the GPFS daemon starts. If the keystore files are stored on an NFS mount, the encryption initialization process can hang. The cause is a bug that affects the way NFS handles record locking. If you encounter this problem, upgrade your version of NFS or store your keystore file on a local file system. If an upgrade is not possible and no local file system is available, use a RAM drive to store the keystore files.

See the following subtopics for instructions:

[“Part 1: Installing Security Key Lifecycle Manager” on page 789](#)

[“Part 2: Creating and exporting a server certificate” on page 791](#)

[“Part 3: Configuring the remote key management \(RKM\) back end” on page 793](#)

[“Part 4: Configuring more RKM back ends” on page 796](#)

Part 1: Installing Security Key Lifecycle Manager

Follow the instructions in this subtopic to install and configure the IBM Security Key Lifecycle Manager (SKLM).

1. Install IBM Security Key Lifecycle Manager. For the supported versions, see “[Preparation for encryption](#)” on page 742. For the installation, choose a system that the IBM Storage Scale node that you want to configure has direct network access to. For information about installing SKLM, see the *Installing and configuring* chapter of the SKLM documentation.
2. From the main page of the SKLM web GUI, click **Configuration > Key Serving Parameters** and select the check box for **Keep pending client device communication certificates**.
3. Configure SKLM to have the same FIPS 140-2 (FIPS) setting as the IBM Storage Scale cluster.

Follow these steps:

- a) Determine the FIPS setting of the cluster by entering the following command on the command line:

```
mmlsconfig FIPS1402mode
```

The command returns yes if the cluster complies with FIPS or no if not.

- b) On the SKLM server system, open the SKLMConfig.properties file.

Note: The default location of the SKLMConfig.properties file depends on the operating system:

- On AIX, Linux, and similar operating systems the directory is at the following location:

- For GKLM 4.1.1 or later versions:

```
/opt/IBM/WebSphere/Liberty/products/sklm/config/SKLMConfig.properties
```

- For GKLM 4.1.0.1 and old supported SKLM versions:

```
/opt/IBM/WebSphere/AppServer/products/sklm/config/  
SKLMConfig.properties
```

- On Microsoft Windows, the directory is at the following location:

- For GKLM 4.1.1 or later versions:

```
Drive:\Program Files
```

```
(x86)\IBM\WebSphere\Liberty\products\sklm\config\SKLMConfig.properties
```

- For GKLM 4.1.0.1 and old supported SKLM versions:

```
Drive:\Program Files
```

```
(x86)\IBM\WebSphere\AppServer\products\sklm\config\SKLMConfig.properties
```

- c) Add or remove the following line from the SKLMConfig.properties file.

Add the line to configure SKLM to comply with FIPS, or remove it to have SKLM not comply with FIPS.

```
fips=on
```

4. Configure the SKLM server to have the same NIST SP800-131a (NIST) setting as the IBM Storage Scale cluster. Follow these steps:

- a) Determine the NIST setting of the cluster by entering the following command on the command line:

```
mmlsconfig nistCompliance
```

The command returns SP800-131A if the cluster complies with NIST or off if it is non-compliant.

- b) On the SKLM server system, open the SKLMConfig.properties file. For the location of this file, see the note in Step 3.

- c) Add the following line to configure SKLM to comply with NIST or remove it to configure SKLM not to comply with NIST:

```
TransportListener.ssl.protocols=TLSv1.2
```

5. If the cipher suites are set at any time, SKLM 2.6.0.0 has a known issue that causes server certificates always to be signed with SHA1withRSA. To work around the problem, follow these steps:

- a) While the SKLM server is running, in the SKLMConfig.properties file, add or modify the requireSHA2Signatures property as follows:

```
requireSHA2Signatures=true
```

- b) Do not restart the server.

- c) Generate a new server certificate and set it to be the one in use.

- d) If you restart the server, you must repeat this workaround before you can create a server certificate that is signed other than with SHA1withRSA.

Part 2: Creating and exporting a server certificate

Follow the instructions in this subtopic to create and export a server certificate in SKLM:

1. Create a self-signed server certificate:
 - a) On the system where SKLM is running, open the graphical user interface.
 - b) Click **Configuration > SSL/KMIP**.
 - c) Click **Create self-signed certificate**.
 - d) Enter the information for the certificate and click **OK**.
 - e) Restart the server to verify that the server can operate with the new certificate.
2. Make a note of the label of the certificate that is in use:
 - a) In the SKLM graphical user interface, click **Advanced Configuration > Server Certificates**.
 - b) Select the certificate that is identified as being in use. Click **Modify** and make a note of the certificate label. You need it in Step 3.
3. Export the certificate through the command-line interface. Follow these steps:
 - a) On the SKLM server system, open a command-line window.
 - b) Change to the *WAS_HOME/bin* directory. The location of this directory depends on the operating system:
 - On AIX, Linux, and similar operating systems, the directory is at the following location:
 - For GKLM 4.1.1 and later versions:
`/opt/IBM/WebSphere/Liberty/bin`
 - For GKLM 4.1.0.1 and old supported SKLM versions:
`/opt/IBM/WebSphere/AppServer/bin`
 - On Microsoft Windows, the directory is at the following location:
 - For GKLM 4.1.1 and later versions:
`drive:\Program Files (x86)\IBM\WebSphere\Liberty\bin`
 - For GKLM 4.1.0.1 and old supported SKLM versions:
`drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin`
 - c) Enter the following command to start the command-line interface to SKLM:
 - On AIX, Linux, and similar operating systems:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```
 - On Microsoft Windows:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```

- d) In the SKLM command line interface, enter the following command:

```
print AdminTask.tklmCertList('[-alias labelSSCert]')
```

where:

labelSSCert

Specifies the certificate label of the self-signed server certificate. You made a note of the label in Step 2.

SKLM responds with output like the following example:

```
CTGKM0001I Command succeeded.  
uuid = CERTIFICATE-7005029a-831d-405f-af30-4bf0177909de  
alias = server  
key store name = defaultKeyStore  
key state = ACTIVE
```

```
issuer name = CN=server
subject name = CN=server
creation date = 13/03/2014 16:27:13 Eastern Daylight Time
expiration date = 09/03/2015 07:12:30 Eastern Daylight Time
serial number = 1394363550
```

- e) Make a note of the UUID of the certificate that is displayed on line 2 of the output. You need it in the next substep and in Part 3.
- f) To export the certificate, from the SKLM command line interface, enter the following command on one line:

```
print AdminTask.tklmCertExport('[-uuid certUUID -format base64 -fileName fileName]')
```

where:

certUUID

Specifies the UUID that you made a note of in the previous substep.

fileName

Specifies the path and file name of the certificate file in which the server certificate is stored.

SKLM exports the self-signed server certificate into the specified file.

- g) Close the SKLM command line interface.
 - h) Copy the server certificate file to a temporary directory on the IBM Storage Scale node that you are configuring for encryption.
4. In SKLM, create a device group and keys for the IBM Storage Scale cluster:
 - a) In the SKLM graphical user interface, click **Advanced Configuration > Device Group**.
 - b) In the **Device Group** table, click **Create**.
 - c) In the **Create Device Group** window, follow these steps:
 - i) Select the **GPFS** device family.
 - ii) Enter an appropriate name, such as GPFS_Tenant0001. The name is case-sensitive.
 - iii) Make a note of the name. You need it in Part 3 when you create an RKM stanza.
 - iv) Complete any other fields and click **Create**.
 - d) After SKLM creates the device group, it prompts you to add devices and keys. Do not add any devices or keys. Instead, click **Close**. Keys are created in the next step.
 5. Create keys for the device group.
 - a) In the SKLM graphical user interface, in the Key and Device Management table, select the device group that you created in Step 4. In these instructions the device group is named GPFS_Tenant0001.
 - b) Click **Go to > Manage keys and services**.
 - c) In the management page for GPFS_Tenant0001, click **Add > Key**.
 - d) Enter the following information:
 - The number of keys to be created
 - The three-letter prefix for key names. The key names are internal SKLM names and are not used for GPFS encryption.
 - e) Make a note of the UUID of the key, such as KEY-326a1906-be46-4983-a63e-29f005fb3a15. You need it in Part 3.
 - f) In the drop-down list at the bottom of the page, select **Hold new certificate requests pending my approval**.

Part 3: Configuring the remote key management (RKM) back end

An RKM back end defines a connection between a local key client, a remote key tenant, and an RKM server. Each RKM back end is described in an RKM stanza in an RKM.conf file on each node that is configured for encryption.

This subtopic describes how to configure a single RKM back end and how to share the configuration among multiple nodes in a cluster. To configure multiple RKM back ends, see “[Part 4: Configuring more RKM back ends](#)” on page 796

You can do Step 1 and Step 2 on any node of the cluster. In later steps, you will copy the configuration files from Step 1 and Step 2 to other nodes in the cluster.

1. Create and configure a client keystore. Follow these steps:

- a) Create the following subdirectory to contain the client keystore:

```
/var/mmfs/etc/RKMcerts
```

- b) The following command creates the client keystore, stores a private key and a client certificate in it, and also stores the trusted SKLM server certificate into it. From the command line, enter the following command on one line:

```
mmauth gencert --cname clientName --cert serverCertFile --out /var/mmfs/etc/RKMcerts/SKLM.p12  
--label clientCertLabel --pwd-file passwordFile
```

where the parameters are as follows:

--cname *clientName*

The name of the client that is used in the certificate.

--cert *serverCertFile*

The path and file name of the file that contains the SKLM server certificate. You extracted this certificate from SKLM and copied the certificate file to the node in Part 2, Step 3(h).

--out */var/mmfs/etc/RKMcerts/SKLM.p12*

The path and file name of the client keystore.

--label *clientCertLabel*

The label of the client certificate in the keystore. The label can be 1 - 20 characters in length.

--pwd-file *passwordFile*

The path of a text file that contains the password for the client keystore. The password can be 1 - 20 characters in length.

Important: Verify that the files in the client keystore directory meet the requirements for security-sensitive files that are listed in the [Requirements](#) section at the beginning of this topic.

2. Create an RKM.conf file and add a stanza to it that describes a connection between a local key client, an SKLM device group, and an SKLM key server. Each stanza defines an RKM back end.

- a) Create a text file with the following path and name:

```
/var/mmfs/etc/RKM.conf
```

Important: Verify that the files in the client keystore directory meet the requirements for security-sensitive files that are listed in the [Requirements](#) section at the beginning of this topic.

- b) Add a stanza with the following format:

```
stanzaName {  
    type = ISKLM  
    kmipServerUri = tls://raclette.zurich.ibm.com:5696  
    keyStore = /var/mmfs/etc/RKMcerts/SKLM.p12  
    passphrase = a_password  
    clientCertLabel = a_label  
    tenantName = GPFS_Tenant0001  
}
```

where the rows of the stanza have the following meaning:

stanzaName

A name (RKM ID) for the stanza. Make a note of the name: you need it in the next step.

It is important to ensure that the RKM ID is unique among all RKM stanzas that are configured and no longer than 21 characters in length.

type

Always ISKLM.

kmipServerUri

The DNS name or IP address of the SKLM server and the KMIP SSL port. You can find this information on the main page of the SKLM graphic user interface. The default port is 5696.

You can have multiple instances of this line, where the first instance represents the primary key server and each additional instance represents a backup key server. You can have up to five backup key servers. The following example has the primary key server and five backup key servers:

```
stanzaName {  
    type = ISKLM  
    kmipServerUri = tls://raclette.zurich.ibm.com:5696  
  
    kmipServerUri2 = tls://raclette.fondue2.ibm.com:5696  
    kmipServerUri3 = tls://raclette.fondue3.ibm.com:5696  
    kmipServerUri4 = tls://raclette.fondue4.ibm.com:5696  
    kmipServerUri5 = tls://raclette.fondue5.ibm.com:5696  
    kmipServerUri6 = tls://raclette.fondue6.ibm.com:5696  
    keyStore = /var/mmfs/etc/RKMcerts/SKLM.p12  
    passphrase = a_password  
    clientCertLabel = a_label  
    tenantName = GPFS_Tenant0001  
}
```

If the GPFS daemon cannot get an encryption key from the primary key server, it tries the backup key servers in order.

For more information, see the subtopics "RKM back ends" and "Adding backup RKM servers in a high-availability configuration" in the topic ["Preparation for encryption" on page 742](#).

keyStore

The path and name of the client keystore. You specified this parameter in Step 1.

passphrase

The password of the client keystore and client certificate. You specified this parameter in Step 1.

clientCertLabel

The label of the client certificate in the client keystore. You specified this parameter in Step 1.

tenantName

The name of the SKLM device group. See ["Part 1: Installing Security Key Lifecycle Manager" on page 789](#).

3. Copy the configuration files to the file system manager node:



Warning: The **mmchpolicy** command in Step 5 fails if you omit this step. The **mmchpolicy** command requires the configuration files to be on the file system manager node.

- a) Copy the RKM.conf file from the /var/mmfs/etc directory to the same directory on the file system manager node.
- b) Copy the keystore files that the RKM file references to the same directories on the file system manager node. The recommended location for the keystore files is /var/mmfs/etc/RKMcerts/.

Important: Verify that the files in the client keystore directory meet the requirements for security-sensitive files that are listed in the Requirements section at the beginning of this topic.

4. To configure other nodes in the cluster for encryption, copy the RKM.conf file and the keystore files to those nodes.

Copy the files in the same way as you did in Step 3.

Important: Verify that the files in the client keystore directory meet the requirements for security-sensitive files that are listed in the Requirements section at the beginning of this topic.

5. Install an encryption policy for the cluster:

Note: You can do this step on any node to which you copied the configuration files.

a) Create a policy that instructs GPFS to do the encryption tasks that you want.

The following policy is an example policy. It instructs IBM Storage Scale to encrypt all files in the file system with a file encryption key (FEK) and to wrap the FEK with a master encryption key (MEK):

```
RULE 'p1' SET POOL 'system' /* one placement rule is required at all times */
RULE 'Encrypt all files in file system with rule E1'
SET ENCRYPTION 'E1'
WHERE NAME LIKE '%'
RULE 'simpleEncRule' ENCRYPTION 'E1' IS
ALGO 'DEFAULTNISTSP800131A'
KEYS('KEY-326a1906-be46-4983-a63e-29f005fb3a15:SKLM_srv')
```

In the last line of the policy, the character string within single quotation marks ('') is the key name. A *key name* is a compound of two parts in the following format:

KeyID:RkmID

where:

KeyID

Specifies the UUID of the key that you created in the SKLM graphic user interface in Part 2.

RkmID

Specifies the name of the RKM backend stanza that you created in the /var/mmf5/etc/RKM.conf file.

b) Install the policy rule with the **mmchpolicy** command.



Trouble: If an encryption policy succeeds on one node but fails on another node in the same cluster, verify that the failing node has the correct client keystore and stanza.



CAUTION: Installing a new policy with the **mmchpolicy** command removes all the statements in the previous policy. To add statements to an existing policy without deleting the previous contents, collect all policy statements for the file system into one file. Add the new statements to the file and install the contents of the file with the **mmchpolicy** command.

6. Import the client certificate into the SKLM server:

a) On the IBM Storage Scale node that you are configuring for encryption, send a KMIP request to SKLM.

To send a KMIP request, try to create an encrypted file on the node. The attempt fails, but it causes SKLM to put the client certificate in a list of pending certificates in the SKLM key server. The attempt fails because SKLM does not yet trust the client certificate. See the following example:

```
# touch /gpfs0/test
touch: cannot touch '/gpfs0/test': Permission denied
# tail -n 2 /var/adm/ras/mmf5.log.latest
Thu Mar 20 14:00:55.029 2014: [E] Unable to open encrypted file: inode 46088,
Fileset fs1, File System gpfs0.
Thu Mar 20 14:00:55.030 2014: [E] Error: key
'KEY-326a1906-be46-4983-a63e-29f005fb3a15:SKLM_srv' could not be fetched (RKM
reported error -1004).
```

b) In the graphical user interface of SKLM, on the main page, click **Pending client device communication certificates**.

c) Find the client certificate in the list and click **View**.

- d) Carefully check that the certificate that you are importing matches the one created in the previous step, then click **Accept and Trust**.
- e) On the resulting screen, provide a name for the certificate and click **Accept and Trust** again.
- f) On the node that you are configuring for encryption, try to create an encrypted file as you did in Step (a).

This time the command succeeds. Enter an `mmlsattr` command to list the encryption attributes of the new file:

```
# touch /gpfs0/test
# mmlsattr -n gpfs.Encryption /gpfs0/test
file name: /gpfs0/test
gpfs.Encryption: "EAGC????f????????????????? ??????w?^??>????????????? ?L4??
-???V}f??X????,?G?<sH??O?)??M?????)?KEY-326a1906-be46-4983-a63e-29f005fb3a15?
sklmsrv?)?KEY-6aaa3451-6a0c-4f2e-9f30-d443ff2ac7db?RKMKMP3?"'
EncPar 'AES:256:XTS:FEK:HMACSHA512'
type: wrapped FEK WrPPar 'AES:KWRAP' CmbPar 'XORHMACSHA512'
KEY-326a1906-be46-4983-a63e-29f005fb3a15:sklmsrv
```

From now on, the encryption policy rule causes each newly created file to be encrypted with a file encryption key (FEK) that is wrapped in a master encryption key (MEK). You created the key in a device group in the SKLM server and included its UUID as part of a key name in the security rule.

Part 4: Configuring more RKM back ends

To configure more RKM back ends, follow the steps in Part 3.

You might want to:

- Add a primary or backup key server.
- Add a key client by creating and configuring a client keystore and importing the client certificate into the SKLM server.
- Define a back end by adding a stanza to the `RLM.conf` file. You can share client keystores, tenants, or key servers between stanzas.

Note the following design points:

- On a single node:
 - The `RKM.conf` file can contain multiple stanzas. Each stanza represents a connection between a key client and an SKLM device group.
 - You can create multiple keystores.
- Across different nodes:
 - The contents of `RKM.conf` files can be different.
 - The contents of keystores can be different.
 - If an encryption policy succeeds on one node and fails on another in the same cluster, verify that the failing node has the correct client keystore and stanza.

Remember: All nodes that mount a file system need to be able to access all the keys used in that file system.

- Add encryption policies. Before you run the `mmchpolicy` command, ensure that the following conditions have been met:
 - The keystore files and the `RKM.conf` files have been copied to the proper nodes.
 - The files in the client keystore directory meet the requirements for security-sensitive files that are listed in the [Requirements](#) section at the beginning of this topic.

For more information, see “[RKM back ends](#)” on page 744 in “[Preparation for encryption](#)” on page 742.

Regular setup: Using SKLM with a certificate chain

Learn to use the regular setup method to configure the key client node with the IBM Security Key Lifecycle Manager (SKLM) key server when the server is running with a certificate chain from a certificate authority (CA) rather than with a self-signed server certificate.



Attention: The simplified setup method, which can be used only when the Remote Key Management (RKM) server is SKLM, is much easier to use and more powerful than the regular setup method with SKLM. In the simplified setup method, the **mmkeyserv** command automatically performs many of the steps that must be done manually in the regular setup method.

The regular setup method with SKLM requires IBM Storage Scale Advanced Edition, IBM Storage Scale Data Management Edition, or IBM Storage Scale Developer Edition or IBM Storage Scale Erasure Code Edition V4.1 or later and a supported version of SKLM. For information about supported SKLM versions, see [“Preparation for encryption” on page 742](#).

This topic describes the regular method for setting up encryption with SKLM as the RKM server and with a certificate that is signed by a certificate authority CA on the KMIP port of the SKLM server. If your deployment scenario uses a self-signed server certificate, see one of the following topics:

[“Simplified setup: Using SKLM with a self-signed certificate” on page 747](#)

[“Regular setup: Using SKLM with a self-signed certificate” on page 788](#)

Note: If you are using SKLM v2.7 or later, see the topic [“Configuring encryption with SKLM 2.7 or later” on page 810](#).

Requirements:

The following requirements must be met on every IBM Storage Scale node that participates in encryption:

- The node must have direct network access to the system where the key server is installed.
- The security-sensitive files that are created during the configuration process must have the following characteristics:
 - They must be regular files that are owned by the root user.
 - The group ownership must be changed to root group.
 - They must be readable and writable only by the user (mode '0600'). The following examples apply to the regular setup with SKLM and with Thales Vormetric Data Security Manager (DSM) setup:

```
-rw----- 1 root root 2446 Mar 20 12:15 /var/mmfs/etc/RKM.conf  
drw----- 2 root root 4096 Mar 20 13:47 /var/mmfs/etc/RKMcerts  
-rw----- 1 root root 3988 Mar 20 13:47 /var/mmfs/etc/RKMcerts/keystore_name.p12
```

These security-sensitive files include the following files:

- The RKM.conf file. For more information about this file, see [“The RKM.conf file and the RKM stanza” on page 744](#).
- The files in the client keystore directory, which include the keystore file, the public and private key files for the client, and possibly other files. For more information about these files, see [“The client keystore directory and its files” on page 746](#).



CAUTION:

- Take appropriate precautions to ensure that the security-sensitive files are not lost or corrupted. IBM Storage Scale does not manage or replicate the files.
- Ensure that the passphrase for the client certificate file is not leaked through other means, such as the shell history.
- Client keystore files must be record-locked when the GPFS daemon starts. If the keystore files are stored on an NFS mount, the encryption initialization process can hang. The cause is a bug that affects the way NFS handles record locking. If you encounter this problem, upgrade your version of NFS or store your keystore file on a local file system. If an upgrade is not possible and no local file system is available, use a RAM drive to store the keystore files.

See the following subtopics for instructions:

- [“Part 1: Installing Security Key Lifecycle Manager” on page 798](#)
- [“Part 2: Configuring SKLM” on page 799](#)
- [“Part 3: Configuring the remote key management \(RKM\) back end” on page 801](#)
- [“Part 4: Enabling encryption on other nodes” on page 806](#)

Part 1: Installing Security Key Lifecycle Manager

Follow the instructions in this subtopic to install and configure the IBM Security Key Lifecycle Manager (SKLM).

1. Install IBM Security Key Lifecycle Manager. For the supported versions, see [“Preparation for encryption” on page 742](#). For the installation, choose a system that the IBM Storage Scale node that you want to configure has direct network access to. For more information, see the *Installing and configuring* chapter of the SKLM documentation.
2. From the main page of the SKLM web GUI, click **Configuration > Key Serving Parameters** and select the checkbox for **Keep pending client device communication certificates**.
3. Configure SKLM to have the same FIPS 140-2 (FIPS) setting as the IBM Storage Scale cluster.

Follow these steps:

- a) Determine the FIPS setting of the cluster by issuing the following command:

```
mmlsconfig FIPS1402mode
```

The command returns yes if the cluster complies with FIPS or no if not.

- b) On the SKLM server system, open the `SKLMConfig.properties` file.

Note: The default location of the `SKLMConfig.properties` file depends on the operating system:

- On AIX, Linux, and similar operating systems the directory is at the following location:
 - For GKLM 4.1.1 or later versions:
`/opt/IBM/WebSphere/Liberty/products/sk1m/config/SKLMConfig.properties`
 - For GKLM 4.1.0.1 and old supported SKLM versions:
`/opt/IBM/WebSphere/AppServer/products/sk1m/config/SKLMConfig.properties`
- On Microsoft Windows, the directory is at the following location:
 - For GKLM 4.1.1 or later versions:
`Drive:\Program Files
(x86)\IBM\WebSphere\Liberty\products\sk1m\config\SKLMConfig.properties`
 - For GKLM 4.1.0.1 and old supported SKLM versions:
`Drive:\Program Files
(x86)\IBM\WebSphere\AppServer\products\sk1m\config\SKLMConfig.properties`

- c) In the `SKLMConfig.properties` file, find the line that begins `fips=`. To configure the FIPS setting for SKLM, enter `fips=on` to comply with FIPS or `fips=off` not to comply. If the line is not present in the file, add it.

4. Configure the SKLM server to have the same NIST SP800-131a (NIST) setting as the IBM Storage Scale cluster. Follow these steps:

- a) Determine the NIST setting of the cluster by issuing the following command on the command line:

```
mmlsconfig nistCompliance
```

The command returns `SP800-131A` if the cluster complies with NIST or `off` if not.

- b) On the SKLM server system, open the `SKLMConfig.properties` file. For the location of this file, see the note in Step 3.
- c) Add the following line to configure SKLM to comply with NIST or remove it to configure SKLM not to comply with NIST:

```
TransportListener.ssl.protocols=TLSv1.2
```

5. Configure IBM WebSphere Application Server so that it has the same NIST setting as the IBM Storage Scale cluster.

See the topic [Transitioning WebSphere Application Server to the SP800-131 security standard](#) in the volume *WebSphere Application Server Network Deployment* in the WebSphere Application Server online documentation.

- WebSphere Application Server can be configured to run SP800-131 in a transition mode or a strict mode. The strict mode is recommended.
 - When NIST is enabled, make sure that WebSphere Application Server certificate size is at least 2048 bytes and is signed with SHA256withRSA as described in the preceding link.
6. If the cipher suites were set at any time, SKLM 2.6.0.0 has a known issue that causes server certificates always to be signed with SHA1withRSA. To work around the problem, follow these steps:

- a) While the SKLM server is running, in the `SKLMConfig.properties` file, modify the `requireSHA2Signatures` property as follows:

```
requireSHA2Signatures=true
```

- b) Do not restart the server.
- c) Generate a new server certificate signing request (CSR) to a third-party certificate authority (CA) and send it to the CA.
- d) When you receive the certificate from the third-party CA, import it into SKLM and set it to be the certificate in use. For more information, see the next subtopic.
- e) If you restart the server, you must repeat this workaround before you can create a server certificate that is signed other than with SHA1withRSA.

Part 2: Configuring SKLM

To configure SKLM, you must create a certificate signing request (CSR), send it to the CA, obtain the certificate chain from the CA, and import the endpoint certificate into the SKLM server. You must also create a device group for the cluster and create keys for the device group.

Note: For more information about the steps in this subtopic, see [Scenario: Request for a third-party certificate](#) in IBM Security Guardium Key Lifecycle Manager documentation.

1. Create a CSR with the SKLM command line interface:
 - a) On the SKLM server system, open a command-line window.
 - b) Change to the `WAS_HOME/bin` directory. The location of this directory depends on the operating system:
 - On AIX, Linux, and similar operating systems, the directory is at the following location:
 - For GKLM 4.1.1 and later versions:
`/opt/IBM/WebSphere/Liberty/bin`
 - For GKLM 4.1.0.1 and old supported SKLM versions:
`/opt/IBM/WebSphere/AppServer/bin`
 - On Microsoft Windows, the directory is at the following location:
 - For GKLM 4.1.1 and later versions:
`drive:\Program Files (x86)\IBM\WebSphere\Liberty\bin`

- For GKLM 4.1.0.1 and old supported SKLM versions:

```
drive:\Program Files (x86)\IBM\WebSphere\AppServer\bin
```

c) Start the command line interface to SKLM:

- On AIX, Linux, and similar operating systems, issue the following command:

```
./wsadmin.sh -username SKLMAdmin -password mypwd -lang jython
```

- On Microsoft Windows, issue the following command:

```
wsadmin -username SKLMAdmin -password mypwd -lang jython
```

d) In the SKLM command line interface, issue the following command on one line:

```
print AdminTask.tkLMCertGenRequest('[-alias labelCsr -cn server  
-validity daysValid -keyStoreName defaultKeyStore -fileName fileName -usage SSLSERVER]')
```

where:

-alias labelCsr

Specifies the certificate label of the CSR.

-cn server

Specifies the common name of the server in the certificate.

-validity daysValid

Specifies the validity period of the certificate in days.

-keyStoreName defaultKeyStore

Specifies the keystore name within SKLM where the CSR is stored. Typically, you would specify defaultKeyStore as the name here.

-fileName fileName

Specifies the fully qualified path of the directory where the CSR is stored on the SKLM server system, for example /root/skLMServer.csr.

-usage SSLSERVER

Specifies how the generated certificate is used in SKLM.

The following example shows the SKLM response:

```
CTGKM0001I Command succeeded  
fileName
```

2. Send the CSR file from Step 1 to the certificate authority.

3. When you receive the generated certificate file, or *endpoint certificate* file, from the certificate authority, copy it to a directory on the node that you are configuring for encryption. For example, you might copy it to the directory and file /opt/IBM/WebSphere/Liberty/products/skLM/data/skLMServer.cert.

Important:

a. You must also obtain and copy the root certificate file and any intermediate certificate files into the same temporary directory. The root certificate and the intermediate certificates might be included with the generated endpoint certificate file. Or you might have to obtain the root certificate file and any intermediate certificate files separately. Whatever the method, you must have a root certificate file, any intermediate certificate files, and the endpoint certificate file. You need these certificate files in Part 3.

b. If you have not already done so, save the files of the certificate chain to a secure location. Include the root certificate file, any intermediate certificate files, and the endpoint certificate file. Now, when a client certificate expires, you will not need to download the certificate chain from the server again. You can add your local copy of the files in the server certificate chain to the new client keystore. For more information, see [“Renewing expired client certificates” on page 845](#).

4. Import the endpoint certificate into the SKLM server with the SKLM graphical user interface:

- a) On the **Welcome** page, in the **Action Items** section, in the **Key Groups and Certificates** area, click **You have pending certificates**.
 - b) In the **Pending Certificates** table, click the certificate that you want to import and click **Import**.
 - c) In the **File name and location** field, type the path and file name of the certificate file and click **Import**.
5. In SKLM, create a device group for the IBM Storage Scale cluster:
- a) In the SKLM graphical user interface, click **Advanced Configuration > Device Group**.
 - b) In the **Device Group** table, click **Create**.
 - c) In the **Create Device Group** window, follow these steps:
 - i) Select the **GPFS** device family.
 - ii) Enter an appropriate name, such as GPFS_Tenant0001. The name is case-sensitive.
 - iii) Make a note of the name. You need it in Part 3 when you create an RKM stanza.
 - iv) Complete any other fields and click **Create**.
 - d) After SKLM creates the device group, it prompts you to add devices and keys. Do not add any devices or keys. Instead, click **Close**. Keys are created in the next step.
6. Create master encryption keys for the device group.
- a) In the SKLM graphical user interface, in the **Key and Device Management** table, select the device group that you created in Step 5. In these instructions, the device group is named GPFS_Tenant0001.
 - b) Click **Go to > Manage keys and services**.
 - c) In the management page for GPFS_Tenant0001, click **Add > Key**.
 - d) Enter the following information:
 - The number of keys to be created.
 - The three-letter prefix for key names. The key names are internal SKLM names and are not used for GPFS encryption.
 - e) Make a note of the UUID of the key, such as KEY-326a1906-be46-4983-a63e-29f005fb3a15. You need it in Part 3.
 - f) In the drop-down list at the bottom of the page, select **Hold new certificate requests pending my approval**.

Part 3: Configuring the remote key management (RKM) back end

To configure a remote key management (RKM) back end, you must create and initialize a client keystore and you must create an RKM stanza in the RKM.conf file on the IBM Storage Scale node:

1. On the IBM Storage Scale node that you are configuring for encryption, create the following subdirectory to contain the client keystore:

```
/var/mmfs/etc/RKMcerts
```

2. Issue the following command to create the client credentials. The command is all on one line:

```
mmgskkm gen --prefix /var/mmfs/etc/RKMcerts/SKLM --cname clientName
--pwd-file passwordFile --fipsVal --nist nistVal --days validDays --keylen keyBits
```

where:

--prefix /var/mmfs/etc/RKMcerts/SKLM

Specifies the path and file name prefix of the client credential files that are generated.

--cname clientName

Specifies the name of the client in the certificate.

--pwd-file passwordFile

Specifies the path of a text file that contains the password for the client keystore. The password must be 1 - 20 characters in length.

--fips fipsVal

Specifies the current FIPS 140-2 compliance mode of the IBM Storage Scale cluster. Valid values are on and off. To find the current mode, issue the following command:

```
mmlsconfig fips1402mode
```

--nist nistVal

Specifies the current NIST SP 800-131A compliance mode of IBM Storage Scale cluster. Valid values are on and off. To find the current mode, issue the following command:

```
mmlsconfig nistCompliance
```

--days validDays

Specifies the number of days that the client certificate is valid.

--keylen keyBits

Specifies the number of bits for the client private RSA key.

The command creates three files that contain the private key, the public key, and the certificate for the client.

3. Issue the following command to create the client keystore and store the private key and the client certificate in it. The command is all on one line:

```
mmgskkm store --cert /var/mmfs/etc/RKMcerts/SKLM.cert  
--priv /var/mmfs/etc/RKMcerts/SKLM.priv --label clientCertLabel  
--pwd-file passwordFile --out /var/mmfs/etc/RKMcerts/SKLM.p12 --fips fipsVal --nist nistVal
```

where:

--cert /var/mmfs/etc/RKMcerts/SKLM.cert

Specifies the path of the client certificate file. The path was specified in the --prefix parameter Step 2. The file suffix is .cert.

--priv /var/mmfs/etc/RKMcerts/SKLM.priv

Specifies the path of the client private key file. The path was specified in the --prefix parameter in the Step 2. The file suffix is .priv.

--label clientCertLabel

Specifies the label of the client certificate in the keystore.

--pwd-file passwordFile

Specifies the path of a text file that contains the password for the client keystore. The password must be 1 - 20 characters in length.

--out /var/mmfs/etc/RKMcerts/SKLM.p12

Specifies the path of the client keystore.

--fips fipsVal

Specifies the current FIPS 140-2 compliance mode of the IBM Storage Scale cluster. Valid values are on and off. To find the current mode, issue the following command:

```
mmlsconfig fips1402mode
```

--nist nistVal

Specifies the current NIST SP 800-131A compliance mode of the IBM Storage Scale cluster. Valid values are on and off. To find the current mode, issue the following command:

```
mmlsconfig nistCompliance
```

4. Copy the certificate files of the server certificate chain from the temporary directory to the directory that contains the client keystore. You gathered these files in Step 3 of Part 2. Rename each certificate file with the same prefix, followed by a numeral that indicates the order of the certificate in the chain,

followed by the file extension .cert. Start the numbering with 0 for the root certificate. For example, if the chain consists of three certificate files and the prefix is sklmChain, rename the files as follows:

```
sklmChain0.cert  
sklmChain1.cert  
sklmChain2.cert
```

If the certificate chain contains more than three certificate files, combine the intermediate files into one certificate file, set the numeral in the name of the combined certificate file to 1, and set the numeral in the name of the endpoint certificate file to 2. For example, suppose that the certificate chain contains four certificate files: sklmChain0.cert, sklmChain1.cert, sklmChain2.cert, and sklmChain3.cert. Modify the certificate files in the following way:

- The sklmChain0.cert file needs no changes.
- Combine sklmChain1.cert and sklmChain2.cert into one file and name it sklmChain1.cert.
- Rename sklmChain3.cert to sklmChain2.cert.

Important: If you have not already done so, save the files of the certificate chain to a secure location. Include the root certificate file, any intermediate certificate files, and the endpoint certificate file. Now, when a client certificate expires, you will not need to download the certificate chain from the server again. You can add your local copy of the files in the server certificate chain to the new client keystore. For more information, see [“Renewing expired client certificates” on page 845](#).

5. Issue the following command to verify the certificate chain. The command is all on one line:

```
openssl verify -CAfile /var/mmfs/etc/RKMcerts/sklmChain0.cert  
-untrusted /var/mmfs/etc/RKMcerts/sklmChain1.cert /var/mmfs/etc/RKMcerts/sklmChain2.cert
```

where:

-CAfile /var/mmfs/etc/RKMcerts/sklmChain0.cert

Specifies the path of the root certificate file.

-untrusted /var/mmfs/etc/RKMcerts/sklmChain1.cert

Specifies the path of the intermediate certificate file. If no intermediate certificates are in the chain, omit this parameter.

/var/mmfs/etc/RKMcerts/sklmChain2.cert

Specifies the path of the endpoint certificate.

If there are only two certificates, omit the **-untrusted** parameter and issue the command as in the following example:

```
openssl verify -CAfile /var/mmfs/etc/RKMcerts/sklmChain0.cert  
/var/mmfs/etc/RKMcerts/sklmChain1.cert
```

6. Issue the following command to store the certificate chain into the client keystore. The command is all on one line:

```
mmgskkm trust --prefix /var/mmfs/etc/RKMcerts/sklmChain  
--pwd-file passwordFile --out /var/mmfs/etc/RKMcerts/SKLM.p12  
--label labelChain --fips fipsVal --nist nistVal
```

where:

--prefix /var/mmfs/etc/RKMcerts/sklmChain

Specifies the path and the file name prefix of the files in the certificate chain. The mmgskkm command trusts all the files that have the specified prefix and a .cert suffix. For example, if the chain consists of three certificates and the prefix is /var/mmfs/etc/RKMcerts/sklmChain, then the command trusts the following certificates:

```
/var/mmfs/etc/RKMcerts/sklmChain0.cert  
/var/mmfs/etc/RKMcerts/sklmChain1.cert  
/var/mmfs/etc/RKMcerts/sklmChain2.cert
```

--pwd-file passwordFile
Specifies the path of a text file that contains the password of the client keystore.

--out /var/mmfs/etc/RKMcerts/SKLM.p12
Specifies the path of the client keystore.

--label labelChain
Specifies the prefix of the label for the server certificate chain in the client keystore.

--fips fipsVal
Specifies the current FIPS 140-2 compliance mode of the IBM Storage Scale cluster. Valid values are on and off. To find the current mode, issue the following command:

```
mmlsconfig fips1402mode
```

--nist nistVal
Specifies the current NIST SP 800-131A compliance mode of the IBM Storage Scale cluster. Valid values are on and off. To find the current mode, issue the following command:

```
mmlsconfig nistCompliance
```

Important: The new keystore must be record-locked when the GPFS daemon starts. If the keystore files are stored on an NFS mount, the encryption initialization process can hang. The cause is a bug that affects the way NFS handles record locking. If you encounter this problem, upgrade your version of NFS or store your keystore file on a local file system. If an upgrade is not possible and no local file system is available, use a RAM drive to store the keystore files.

7. Create an RKM.conf file and add a stanza to it that contains the information that is necessary to connect to the SKLM key server. The RKM.conf file must contain a stanza for each connection between a key client, an SKLM device group, and a key server.

- In a text editor, create a new text file with the following path and name:

```
/var/mmfs/etc/RKM.conf
```

- Add a stanza with the following format:

```
stanzaName {  
    type = ISKLM  
    kmipServerUri = tls://raclette.zurich.ibm.com:5696  
    keyStore = /var/mmfs/etc/RKMcerts/SKLM.p12  
    passphrase = a_password  
    clientCertLabel = a_label  
    tenantName = GPFS_Tenant0001  
}
```

where the rows of the stanza have the following meaning:

stanzaName

A name (RKM ID) for the stanza. Make a note of the name: you need it in the next step.

It is a good practice to use a format like the following one to ensure that the RKM ID is unique:

```
keyServerName_tenantName
```

where *tenantName* is the name that you provide in the last line of stanza. For example, the RKM ID for the key server and key client in these instructions is: `raclette_GPFS_Tenant0001`.

type

Always ISKLM.

kmipServerUri

The DNS name or IP address of the SKLM server and the KMIP SSL port. You can find this information on the main page of the SKLM graphic user interface. The default port is 5696.

You can have multiple instances of this line, where each instance represents a different backup key server. The following example has the primary key server and two backup key servers:

```

stanzaName {
    type = ISKLM
    kmipServerUri = tls://raclette.zurich.ibm.com:5696
    kmipServerUri = tls://raclette.fondue.ibm.com:5696
    kmipServerUri = tls://raclette.fondue2.ibm.com:5696
    keyStore = /var/mmfs/etc/RKMcerts/SKLM.p12
    passphrase = a_password
    clientCertLabel = a_label
    tenantName = GPFS_Tenant0001
}

```

If the GPFS daemon cannot get an encryption key from the primary key server, it tries the backup key servers in order.

keyStore

The path and name of the client keystore.

passphrase

The password of the client keystore and client certificate.

clientCertLabel

The label of the client certificate in the client keystore.

tenantName

The name of the SKLM device group. See [“Part 1: Installing Security Key Lifecycle Manager” on page 798](#).

8. Set up an encryption policy on the node that you are configuring for encryption.

a) Create a file management policy that instructs GPFS to do the encryption tasks that you want.

The following policy is an example. It instructs IBM Storage Scale to encrypt all files in the file system with a file encryption key (FEK) and to wrap the FEK with a master encryption key (MEK):

```

RULE 'p1' SET POOL 'system' /* one placement rule is required at all times */
RULE 'Encrypt all files in file system with rule E1'
SET ENCRYPTION 'E1'
WHERE NAME LIKE '%'
RULE 'simpleEncRule' ENCRYPTION 'E1' IS
ALGO 'DEFAULTNISTSP800131A'
KEYS('KEY-326a1906-be46-4983-a63e-29f005fb3a15:SKLM_srv')

```

In the last line of the policy, the character string within single quotation marks ('') is the key name. A *key name* is a compound of two parts in the following format:

KeyID:RkmID

where:

KeyID

Specifies the UUID of the key that you created in the SKLM graphic user interface in Part 2.

RkmID

Specifies the name of the RKM backend stanza that you created in the /var/mmfs/etc/RKM.conf file.

b) Issue the **mmchpolicy** command to install the rule.



CAUTION: Installing a new policy with the **mmchpolicy** command removes all the statements in the previous policy. To add statements to an existing policy without deleting the previous contents, collect all policy statements for the file system into one file. Add the new statements to the file and install the contents of the file with the **mmchpolicy** command.

9. Import the client certificate into the SKLM server:

a) On the IBM Storage Scale node that you are configuring for encryption, send a KMIP request to SKLM.

To send a KMIP request, try to create an encrypted file on the node. The attempt fails, but it causes SKLM to put the client certificate in a list of pending certificates in the SKLM key server. The attempt fails because SKLM does not yet trust the client certificate. See the following example:

```

# touch /gpfs0/test
touch: cannot touch '/gpfs0/test': Permission denied
# tail -n 2 /var/adm/ras/mmfs.log.latest
Thu Mar 20 14:00:55.029 2014: [E] Unable to open encrypted file: inode 46088,
Fileset fs1, File System gpfs0.
Thu Mar 20 14:00:55.030 2014: [E] Error: key
'KEY-326a1906-be46-4983-a63e-29f005fb3a15:SKLM_srv' could not be fetched (RKM
reported error -1004).

```

- b) In the graphical user interface of SKLM, on the main page, click **Pending client device communication certificates**.
- c) Find the client certificate in the list and click **View**.
- d) Carefully check that the certificate that you are importing matches the one created in the previous step, then click **Accept and Trust**.
- e) On the resulting screen, provide a name for the certificate and click **Accept and Trust** again.
- f) On the node that you are configuring for encryption, try to create an encrypted file as you did in Step (a).

This time the command succeeds. Issue an `mmlsattr` command to list the encryption attributes of the new file:

```

# touch /gpfs0/test
# mmlsattr -n gpfs.Encryption /gpfs0/test
file name: /gpfs0/test
gpfs.Encryption: "EAGC????f????????????? ?????w?^??>????????????? ?L4??
-???V}f?????????,?G?<H??0?)??M????)?KEY-326a1906-be46-4983-a63e-29f005fb3a15?
sklmsrv?)KEY-6aaa3451-6a0c-4f2e-9f30-d443ff2ac7db?RKMKMP3?"
EncPar 'AES:256:XTS:FEK:HMACSHA512'
type: wrapped FEK WrpPar 'AES:KWRAP' CmbPar 'XORHMACSHA512'
KEY-326a1906-be46-4983-a63e-29f005fb3a15:sklmsrv

```

From now on, the encryption policy rule causes each newly created file to be encrypted with a file encryption key (FEK) that is wrapped in a master encryption key (MEK). You created the key in a device group in the SKLM server and included its UUID as part of a key name in the security rule.

Important: See the security note and the caution at the beginning of this topic before Part 1.

Part 4: Enabling encryption on other nodes

1. To replicate an encryption configuration on another node, you must copy some configuration files from the configured node to the target node:
 - a) Copy the `/var/mmfs/etc/RKM.conf` file to the same directory on the target node.
 - b) Copy the keystore files that the RKM file references to the same directories on the target node. The suggested location for the keystore files on the configured node is `/var/mmfs/etc/RKMcerts/`.
2. To create a different encryption configuration on another node, follow the steps that are described in the preceding subtopics. Note the following design points:
 - On a single node, the following conditions are true:
 - The RKM.conf file can contain multiple stanzas. Each stanza represents a connection between a key client and an SKLM device group.
 - You can create multiple keystores.
 - Across different nodes, the following conditions are true:
 - The contents of RKM.conf files can be different.
 - The contents of keystores can be different.
 - If an encryption policy succeeds on one node and fails on another in the same cluster, verify that the failing node has the correct client keystore and stanza.

Remember: All nodes that mount a file system need to be able to access all the keys used in that file system.

Regular setup: Accessing a remote file system

All nodes that mount a file system must have access all the keys used in the file system. The topic describes steps to configure a remote cluster to mount an encrypted file system when the regular setup is used to configure encryption on the home cluster.

To replicate an encryption configuration on a remote cluster, you must copy encryption configuration files from the configured node in the home cluster to all nodes in the remote cluster.

To copy the Remote Key Management (RKM) server configuration file and the client keystore files on a remote cluster, complete the following steps:

1. If the remote cluster does not have the encryption configuration for other file system, copy the `/var/mmfs/etc/RKM.conf` file into the same directory on the remote nodes.
2. If the remote cluster is configured with regular setup for other file systems, complete the following steps:
 - a) Back up the `/var/mmfs/etc/RKM.conf` file on all nodes in the remote cluster.
 - b) On a single node in the remote cluster, edit the `/var/mmfs/etc/RKM.conf` file to add the RKM stanza that is needed to mount the file system.
 - c) Copy the edited `/var/mmfs/etc/RKM.conf` file into all nodes in the remote cluster.
3. Copy the keystore files that the new RKM stanza references to the same directories on the target node. The suggested location for the keystore files on the configured node is `/var/mmfs/etc/RKMcerts/`.

Converting encryption configuration from regular setup to simplified setup

You can convert an existing IBM Storage Scale encryption configuration that was previously configured using the regular setup (SKLM) to use the simplified setup.

The simplified setup method is much easier to use and more powerful than the regular setup method with Security Key Lifecycle Manager (SKLM). In the simplified setup method, the `mmkeyserv` command automatically performs many of the steps that must be done manually in the regular setup method. The simplified setup method can be used for all supported versions of IBM Storage Scale and SKLM. For more information about the simplified and regular setup, see [“Establishing an encryption-enabled environment” on page 747](#).

Perform the following steps to convert from regular setup to simplified setup:

Note: Steps 1, 2, 3, 4 and 5.a, 5.b can be done while in production, and steps 5.c and 5.d must be done during a maintenance window.

1. Gather information about the current encryption configuration from the `/var/mmfs/etc/RKM.conf` file. The following details are required:
 - Remote Key Manager (RKM) server name
 - Backup servers
 - RKM stanza ID
 - Client name
 - Tenant name

For more information about these terminologies, see [“Preparation for encryption” on page 742](#).

A sample content from the `RKM.conf` file is as follows:

```
SKLM1 {  
    type = ISKLM  
    kmipServerUri = tls://KeyServer01:5696  
    kmipServerUri1 = tls://KeyServer02:5696  
    keyStore = /var/mmfs/etc/RKMcerts/c1Client.p12  
    passphrase = passw0rd  
    clientCertLabel = c1Client  
    tenantName = sklmTenant  
}
```

Where:

- *SKLM1* is the RKM stanza ID.
- *KeyServer01* is the primary RKM server.
- *KeyServer02* is the backup RKM server.
- *c1Client* is the key client name.
- *sklmTenant* is the tenant name.

2. Add the primary and backup RKM servers to the encryption configuration by using the **mmkeyserv** command:

- a) If SKLM is using a self-signed certificate, use the **mmkeyserv server add** command to add the SKLM server to the encryption configuration. Depending on how SKLM is configured, you might also need to specify a port number for connecting with SKLM:

```
# mmkeyserv server add KeyServer01 --backup KeyServer02
```

Where:

- *KeyServer01* is the hostname or IP address of the primary SKLM key server.
- *KeyServer02* is the hostname or IP address of the backup SKLM key server.

- b) If SKLM is using a certificate chain that is signed by a certificate authority, use the **mmkeyserv server add** command with the **--kmpip-cert** option to add the SKLM server to the encryption configuration. Depending on how SKLM is configured, you might also need to specify a port number for connecting with SKLM:

```
mmkeyserv server add KeyServer01 --kmpip-cert CertFilesPrefix --backup KeyServer02
```

Where:

- *KeyServer01* is the hostname or IP address of the primary SKLM key server.
- *KeyServer02* is the hostname or IP address of the backup SKLM key server.
- *CertFilesPrefix* is the path and the file name prefix of the files in the certificate chain.

- c) Issue the **mmkeyserv server show** command to verify that the key server was added successfully. The following example shows that *keyserver01* is added to the encryption configuration:

```
# mmkeyserv server show
keyserver01
  Type: ISKLM
  Hostname: keyserver01.gpfs.net
  User ID: SKLMAdmin
  REST port: 9080
  Label: 1_keyserver01
  NIST: on
  FIPS1402: off
  Backup Key Servers: keyserver02
  Distribute: yes
  Retrieval Timeout: 120
  Retrieval Retry: 3
  Retrieval Interval: 10000
  REST Certificate Expiration: 2033-05-18 17:01:24 (-0400)
  KMIP Certificate Expiration: 2024-02-22 22:24:54 (-0400)
```

3. Add the tenant to the encryption configuration:

- a) Issue the **mmkeyserv** command as shown in the following example to add a tenant:

```
# mmkeyserv tenant add sklmTenant --server keyserver01
```

Where:

- *sklmTenant* is the tenant or device group name. Retrieve the tenant name from the **tenantName** parameter in the /var/mmfs/etc/RKM.conf file.

- *keyserver01* is the hostname or IP address of the primary SKLM key server.
- b) Issue the **mmkeyserv tenant show** command to verify that the tenant was added successfully. The following example shows that the tenant *sklmTenant* is added to *keyserver01*:

```
# mmkeyserv tenant show
sklmTenant
  Key Server: keyserver01.gpfs.net
  Registered Client: (none)
```

4. Create a key client. A key client can request master encryption keys from a tenant after it is registered to the tenant. The **mmkeyserv client create** command creates a client keystore file, which contains client credentials and the certificate of the key server. The command propagates the keystore to all the nodes in the cluster:

- a) Issue the following command to create key client *c1Client1* for key server *keyserver01*. Enter the SKLM administrator password and a passphrase for the new keystore when prompted:

```
# mmkeyserv client create c1Client1 --server keyserver01
Enter password for the key server keyserver01:
Create a pass phrase for keystore:
Confirm your pass phrase:
```

Where:

- *c1Client1* is the key client name. Retrieve the client name from the **clientCertLabel** parameter in the /var/mmfs/etc/RKM.conf file or specify a new key client name.
- *keyserver01* is the host name or IP address of the primary SKLM key server.

- b) Issue the **mmkeyserv client show** command to verify that the key client was created successfully. The following example shows that the key client *c1Client1* is created for remote server *keyserver01*:

```
# mmkeyserv client show
c1Client1
  Label:          c1Client1
  Key Server:    keyserver01.gpfs.net
  Tenants:        (none)
  Certificate Expiration: 2024-02-22 00:01:03 (-0500)
```

5. Register the key client with the tenant to allow the key client to access encryption key in the tenant.

To register a client, you must provide a Remote Key Management (RKM) ID. The RKM ID becomes the identifier field of a new RKM stanza that describes the connection between the key client, the tenant, and the key server. In this step, the key client is registered with a temporary RKM ID that will be changed later to match the RKM ID that is used by the regular setup:

- a) Issue the **mmkeyserv client register** command to register the key client with the tenant using a temporary RKM ID. Enter the requested information when prompted:

```
# mmkeyserv client register c1Client1 --tenant sklmTenant --rkm-id SKLM1temp
Enter password for the key server:

mmkeyserv: [I] Client currently does not have access to the key. Continue the
registration process ...
mmkeyserv: Successfully accepted client certificate
```

- b) Issue the **mmkeyserv rkm show** command to verify that the key client is registered to the tenant:

```
# mmkeyserv rkm show
SKLM1temp {
  type = ISKLM
  kmipServerUri = tls://KeyServer01:5696
  kmipServerUri1 = tls://KeyServer02:5696
  keyStore = /var/mmfs/ssl/keyServ/serverKmip.1_keyserver01.c1Client1.1.p12
  passphrase = fd1gd{XlwXj.X<zEoA
  clientCertLabel = c1Client1
  tenantName = sklmTenant
}
```

Note: The RKM ID must be unique. Therefore, the regular setup RKM configuration file (/var/mmfs/etc/RKM.conf) and the simplified setup RKM configuration file that is managed by **mmkeyserv** command cannot have RKM stanzas with duplicate RKM IDs. In the next step, the RKM ID in the regular setup RKM configuration file must be renamed, commented, or removed before changing the temporary RKM ID that is used to register the key client in the previous step to match the RKM ID used by the regular setup and the file system encryption policy. If an operation that requires reading the RKM stanza from the RKM configuration file is run against the encrypted file system after commenting or removing the RKM ID, but before changing the temporary RKM ID, then the operation might fail due to the inability to find the remote key information. Operations that need to parse the RKM.conf file include mounting the file system and installing a policy.

While the **mmkeyserv rkm change** command typically runs fast, it is recommended to run the RKM ID change operation during a maintenance window. Steps 5.c and 5.d must be run in quick succession.

- c) On all nodes, rename the RKM ID that is being moved to the simplified setup in the regular setup RKM configuration file (/var/mmfs/etc/RKM.conf) or remove the corresponding RKM stanza.
- d) Issue the **mmkeyserv rkm change** command to change the temporary RKM ID (created in step 5.a) to the RKM ID that was used in the regular setup. Enter the requested information when prompted:

```
# mmkeyserv rkm change SKLM1temp --rkm-id SKLM1
mmkeyserv: Propagating the cluster configuration data to all
          affected nodes. This is an asynchronous process.
```

- e) Issue the command **mmkeyserv rkm show** commands as shown in the following example to verify that the RKM ID is changed:

```
# mmkeyserv rkm show
SKLM1 {
type = ISKLM
kmipServerUri = tls://KeyServer01:5696
kmipServerUri1 = tls://KeyServer02:5696
keyStore = /var/mmfs/ssl/keyServ/serverKmip.1_keyserver01.c1Client1.1.p12
passphrase = fd1gd{xlwXj.X<zE0aF
clientCertLabel = c1Client1
tenantName = sklmTenant
}
```

6. Upon successful RKM ID change, encryption is configured with the simplified setup. Repeat the previous steps for each RKM stanza in the regular setup to fully migrate the encryption configuration to the simplified setup.
7. The regular setup configuration files (/var/mmfs/etc/RKM.conf and the old client keystore) can be archived or removed from all nodes in the cluster.

Configuring encryption with SKLM 2.7 or later

Learn to do tasks that are required for the Security Key Lifecycle Manager (SKLM) server 2.7 or later.

Simplified setup: Updating the REST port after upgrading SKLM

Note: IBM Storage Scale supports IBM Security Guardium Key Lifecycle Manager (GKLM) 4.1.0.1 (IF01), 4.1.1, or later. The older versions of GKLM are referred to as IBM Security Lifecycle Manager or SKLM in the documentation. The configuration information is the same for both GKLM and SKLM.

When SKLM is upgraded to a later version, the system administrator might configure the REST interface port of SKLM to a different value. If so, and if the IBM Storage Scale cluster is configured with the simplified setup, you can run the **mmkeyserv server update** command to connect the key client to the new REST interface port. If the new port is not the default REST interface port, you must also specify the new port number in the **--port** parameter of the **mmkeyserv server update** command. For more information, see *mmkeyserv command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

For example, when SKLM is upgraded from 2.6 to 2.7, the REST interface port might be changed from the 2.6 default port (port 9080) to the 2.7 default port (port 443). If this situation occurs, you can run the **mmkeyserv server update** command to connect the IBM Storage Scale key client to the new REST interface port.

Resolving the UUID length problem in IBM Storage Scale versions earlier than 4.2.3

A UUID-length problem arises if a key client that is running a version of IBM Storage Scale earlier than 4.2.3 connects with SKLM version 2.7 or later as the key server. IBM Storage Scale versions earlier than 4.2.3 support a maximum length of 42 characters for the Universally Unique Identifier (UUID) of an encryption key. However, SKLM versions 2.7 and later generate UUIDs of up to 48 characters in length, including a 7 - 8 character Instance ID. To work around this problem, you can configure the SKLM 2.7 or later key server to use one-character instance IDs. After the configuration, the server generates UUIDs that have a maximum length of 42 characters. This method does not change existing UUIDs.

Note:

- IBM Storage Scale supports a maximum key UUID length of 65 characters in versions 4.2.3 and later.
- The instructions in this subsection apply only to versions of IBM Storage Scale earlier than 4.2.3. Do not follow these steps with later versions of IBM Storage Scale.

To configure an SKLM 2.7 or later key server to generate UUIDs with a maximum length of 42 characters, follow these steps:

1. Stop the SKLM server.
2. From the command line, change to the DB2/bin directory.

Note: The location of the DB2/bin directory depends on the operating system:

- On AIX, Linux, and similar operating systems, the directory is at the following location:
`/opt/IBM/DB2SKLMV27/bin`
- On Microsoft Windows, the directory is at the following location:
`Drive:\Program Files\IBM\DB2SKLMV27\bin`

If SKLM uses a preexisting DB2 installation, then the location of the bin directory might be different and might be on another system.

3. Start the DB2 command-line tool. The method depends on the operating system:

- On AIX, Linux, and similar operating systems, enter the following command:
`. /db2`
- On Microsoft Windows, enter the following command:
`db2`

4. At the db2 command-line prompt, enter the following command to list the database directory:

```
list database directory
```

DB2 displays output like the following example:

```
System Database Directory
Number of entries in the directory = 1

Database 1 entry:

Database alias          = SKLMDB27
Database name           = SKLMDB27
Local database directory = /home/sklmdb27
Database release level  = 14.00
Comment                 =
Directory entry type    = Indirect
Catalog database partition number = 0
```

```
Alternate server hostname      =
Alternate server port number   =
```

Make a note of the database name.

- Enter the following command to connect to the SKLM database:

```
connect to database user userName using password
```

Where:

database

Specifies the database name from the previous step.

userName

Specifies the SKLM DB2 user name that you set during SKLM installation. The default value is sk1mdb27.

password

Specifies the SKLM DB2 password that you set during SKLM installation.

- Enter the following command to change the SKLM instance ID. The command is on one line:

```
update KMT_CFGT_INSTDETAILS set INSTANCEID='1' where INSTANCEID in
(select INSTANCEID from KMT_CFGT_INSTDETAILS)
```

where 1 is the one-character Instance ID that you want to set. DB2 displays output like the following example:

```
DB20000I The SQL command completed successfully.
```

- Enter the following command to commit the change:

```
commit
```

DB2 displays output like the following example:

```
DB20000I The SQL command completed successfully.
```

- Enter the following command to close the DB2 command-line tool:

```
quit
```

- Start the SKLM system.

The SKLM key server now generates UUIDs that have a maximum length of 42 characters.

Configuring encryption with the Thales Vormetric DSM key server

This topic describes the regular setup for encryption with Thales Vormetric Data Security Manager (DSM) as the key management server and using self-signed certificates on the KMIP port of the DSM server.

Setting up an encryption environment with DSM as the key server requires IBM Storage Scale Advanced Edition 4.2.1 or later and a supported version of DSM. For more information see the subtopic "Required software: Remote Key Management (RKM) server" in the help topic "[Preparation for encryption](#)" on page 742.

Requirements:

The following requirements must be met on every IBM Storage Scale node that you configure for encryption:

- The node must have direct network access to the system where the key server is installed.
- The security-sensitive files that are created during the configuration process must have the following characteristics:
 - They must be regular files that are owned by the root user.
 - They must be in the root group.

- They must be readable and writable only by the user (mode '0600'). The following examples apply to the regular setup and the DSM setup:

```
-rw----- 1 root root 2446 Mar 20 12:15 /var/mmfs/etc/RKM.conf
drw----- 2 root root 4096 Mar 20 13:47 /var/mmfs/etc/RKMcerts
-rw----- 1 root root 3988 Mar 20 13:47 /var/mmfs/etc/RKMcerts/keystore_name.p12
```

The security-sensitive files include the following files:

- The RKM.conf file. For more information about this file, see “[The RKM.conf file and the RKM stanza](#)” on page 744.
- The files in the client keystore directory, which include the keystore file, the public and private key files for the client, and possibly other files. For more information about these files, see “[The client keystore directory and its files](#)” on page 746.



CAUTION:

- Take appropriate precautions to ensure that the security-sensitive files are not lost or corrupted. IBM Storage Scale does not manage or replicate the files.
- Ensure that the passphrase for the client certificate file is not leaked through other means, such as the shell history.
- Client keystore files must be record-locked when the GPFS daemon starts. If the keystore files are stored on an NFS mount, the encryption initialization process can hang. The cause is a bug that affects the way NFS handles record locking. If you encounter this problem, upgrade your version of NFS or store your keystore file on a local file system. If an upgrade is not possible and no local file system is available, use a RAM drive to store the keystore files.

See the following subtopics for instructions:

- [“Part 1: Creating credentials for the key client” on page 813](#)
- [“Part 2: Configuring the DSM key server” on page 817](#)
- [“Part 3: Configuring the IBM Storage Scale node” on page 819](#)

Part 1: Creating credentials for the key client

- Some of the commands in the following instructions require you to specify values for the following two parameters:

--fips

Specifies whether the key client complies with the requirements of FIPS 140-2.

--nist

Specifies whether security transport for the key client complies with the NIST SP800-131A recommendations.

For both parameters, follow these guidelines:

- If the key client complies, set the parameter to on; otherwise, set the parameter to off.
- Specify the same setting for each parameter as the setting in the IBM Storage Scale cluster. To display these settings, issue the following two commands:

```
mmclsconfig nistCompliance
mmclsconfig FIPS1402mode
```

Follow the procedure shown. If you are using certificates for the client that are signed by a certificate authority (CA), skip Step 1 and go to Step 2.

1. On the IBM Storage Scale node that you are configuring for encryption, run the mmgskkm command to create the client credentials. Issue the following command on one line:

```
/usr/lpp/mmfs/bin/mmgsskkm gen --prefix prefix --cname cname --pwd pwd --fips fips --nist nist
--days valid_days --keylen keylen
```

where:

--prefix *prefix*

Specifies the path and file name prefix of the directory where the output files are generated. For example, if you want directory /var/mmfs/etc/RKMcerts to contain the output files, and you want the output files to have the prefix kcVormetric, you can specify the parameter as follows:

```
--prefix /var/mmfs/etc/RKMcerts/kcVormetric
```

--cname *cname*

Specifies the name of the IBM Storage Scale key client. Valid characters are alphanumeric characters, hyphen (-), and period (.). The name can be up to 54 characters long. In DSM, names are not case-sensitive but avoid the use of uppercase letters. For more information, see the DSM documentation.

--pwd *pwd*

Specifies the password for the private key that this command creates.

--fips *fips*

Specifies whether the key client complies with FIPS 140-2. Specify on or off.

--nist *nist*

Specifies whether the key client complies with NIST SP800-131a. Specify on or off.

--validdays *validdays*

Specifies the number of days that the client certificate is valid.

--keylen *keylen*

Specifies the length in bits of the RSA key that is generated.

In the following example, the current directory is the output directory. Enter the command on one line:

```
/usr/lpp/mmfs/bin/mmgskkm gen --prefix kcVormetric --cname kcVormetric --pwd pwpkVormetric  
--fips off --nist on --days 180 --keylen 2048
```

The output files are a client certificate, a private key, and a public key. For example,

```
kcVormetric.cert  
kcVormetric.priv  
kcVormetric.pub
```

2. Issue the mmgskkm command to create a PKCS#12 keystore and to store the certificate and private key of the client in it.

Note: The input files must follow the specified format as shown in the following list:

- The certificates and private key must be in PEM base64 encoded format.
- The client private key must be unencrypted.
- The chain file must contain the CA root certificate, one or more CA intermediate certificates, and the CA end-point certificate. It must not contain the client certificate.

If you are using the certificate and private file from Step 1, issue the command with the following parameters:

```
/usr/lpp/mmfs/bin/mmgskkm store --cert certFile --priv privFile --label label  
--pwd pwd --out keystore
```

If you are using a certificate chain that is signed by a CA and the certificates are concatenated in a single file, issue the command with the following parameters:

```
/usr/lpp/mmfs/bin/mmgskkm store --cert certFile --priv privFile --chain CACertChainFile  
--label label --pwd pwd --out keystore
```

If you are using a certificate signed by a CA and the certificates are in separate files with the same file prefix, issue the command with the following parameters:

```
/usr/lpp/mmfs/bin/mmgskkm store --cert certFile --priv privFile --prefix CACertFilesPrefix  
--label label --pwd pwd --out keystore
```

The parameters have the same meanings across all three forms of the command:

--cert certFile

Specifies the client certificate file that you created in Step 1 or the client certificate file that is signed by the CA.

--priv privFile

Specifies the private key file that you created in Step 1 or that matches the client certificate that is signed by the CA.

--chain CACertChainFile

Specifies the CA certificate chain file, which contains the CA certificate chain that was used to sign the client certificate. The chain starts with the CA root certificate, continues with any intermediate CA certificates in order, and ends with CA certificate that signed the client certificate. All the certificates are in base64 encoded PEM format. On UNIX-like systems, you can create such a file by concatenating the CA certificates that you received or downloaded from the CA into a single file with the cat command.

--prefix CACertFilesPrefix

Specifies the full path prefix of the CA certificates that are used to sign the client certificate. The CA certificate files must have the format <CACertFilesPrefix><n>.cert, where *CACertFilesPrefix* is the full path prefix for the CA certificate files, such as /tmp/CA/certfiles, and <n> is a CA certificate index. The index is 0 for the CA root certificate and n - 1 for the last intermediate CA certificate that signed the client certificate.

--label label

Specifies the label under which the private key is stored in the keystore.

--pwd pwd

Specifies the password of the keystore. You can use the same password that you specified for the private key in Step 1.

--out keystore

The file name of the keystore.

The output of the command is a client keystore that contains the private key of the client and the certificate or certificate chain of the client.

In the following example, the current directory contains the client credentials from Step 1. The command is on one line:

```
mmgskkm store --cert kcVormetric.cert --priv kcVormetric.priv --label lapkVormetric  
--pwd pwpkVormetric --out ksVormetric.keystore
```

In the following example, the current directory contains the client certificate, the private key, and the client certificate chain. The command is on one line:

```
mmgskkm store --cert kcVormetric.cert --priv kcVormetric.key --chain CACertChain.pem --label  
lapkVormetric  
--pwd pwpkVormetric --out ksVormetric.keystore
```

In the following example, the current directory contains the client certificate, the private key file, and the files in the certificate chain. The certificate files have the format /tmp/CACert<n>.cert, where <n> is the index of the CA certificate in the certificate chain, starting with 0 for the CA root certificate. The command is on one line:

```
mmgskkm store --cert kcVormetric.cert --priv kcVormetric.key --prefix /tmp/CACert --label  
lapkVormetric  
--pwd pwpkVormetric --out ksVormetric.keystore
```

In all three examples, the output file is the client keystore ksVormetric.keystore, which contains the client credentials.

Important: The keystore must be record-locked when the GPFS daemon starts. If the keystore files are stored on an NFS mount, the encryption initialization process can hang. The cause is a bug that affects the way NFS handles record locking. If you encounter this problem, upgrade your version of

NFS or store your keystore file on a local file system. If an upgrade is not possible and no local file system is available, use a RAM drive to store the keystore files.

3. Retrieve the certificate chain of the DSM server.

Note: Before you can do this next step, you must install the DSM server, set up the DSM networking configuration, and set up the server certificate. If you do not, then you might not be able to connect to the DSM server or you might retrieve an invalid, default certificate chain.

Note: DSM does not support the use of imported server certificate chains for the TLS communication on the KMIP port. You must create and use a server certificate chain that is signed by the DSM internal certificate authority (CA).

Enter the following command on one line:

```
/usr/lpp/mmfs/bin/mmsklmconfig restcert --host host --port port --prefix prefix --keystore  
keystore  
--keypass keypass --fips fips --nist nist
```

where:

--host host

Specifies the name or IP address of the remote system where the DSM server is running.

--port port

Specifies the port on the remote system for communicating with the DSM server (default 8445).

--prefix prefix

Specifies the path and file name prefix of the directory where the files in the certificate chain are stored. For example, if you want to store the certificate chain in the directory /var/mmfs/etc/RKMcerts, and you want the certificate files to have the prefix DSMServer, you can specify the parameter as follows:

```
--prefix /var/mmfs/etc/RKMcerts/DSMServer
```

--keystore keystore

Specifies the path and file name of the client keystore that you created in Step 2.

--keypass keypass

Specifies a text file that contains the password of the client keystore as the first line. You must create this text file. Store the password that you provided in Step 2.

--fips fips

Specifies whether the key client complies with FIPS 140-2. Specify on or off.

--nist nist

Specifies whether the key client complies with NIST SP800-131a. Specify on or off.

In the following example, the current directory contains the client keystore that was created in Step 2. Enter the command on one line:

```
/usr/lpp/mmfs/bin/mmsklmconfig restcert --host hostVormetric --port 8445 --prefix DSM  
--keystore ksVormetric.keystore --keypass keypass --fips off --nist on
```

The command connects to the DSM server, retrieves the server certificate chain, and stores each certificate into a separate local file in Base64-encoded DER format. Each file name has the format *prefixN.cert*, where *prefix* is the prefix that you specified in the command and *N* is a digit that begins at 0 and increases by 1 for each certificate in the chain, as in the following example:

DSM0.cert
DSM1.cert

4. Verify that the SHA-256 fingerprint in each retrieved certificate matches the fingerprint of the DSM server:

- To display the details of each certificate, enter the following sequence at the client command line, where *prefix* is the prefix that you provided in Step 3:

```
for c in prefix*.cert; do /usr/lpp/mmfs/bin/mmgskkm print --cert $c; done
```

- b) Log in to the graphical user interface of the DSM server and display its SHA-256 fingerprint.
- c) Verify that the fingerprints in the certificates match the fingerprint in the DSM server.

5. Add the certificates to the PKCS#12 keystore of the key client as trusted certificates. Enter the following command on one line:

```
/usr/lpp/mmfs/bin/mmgskkm trust --prefix prefix --pwd pwd --out keystore --label serverLabel  
--fips fips --nist nist
```

where:

--prefix *prefix*

Specifies the prefix that you specified in Step 3.

--pwd *pwd*

Specifies the password of the client keystore, which you provided in Step 3.

--out *keystore*

Specifies the path name of the keystore of the key client.

--label *serverLabel*

Specifies the label under which the server certificate chain is stored in the client keystore.

--fips *fips*

Specifies whether the key client complies with FIPS 140-2. Specify on or off.

--nist *nist*

Specifies whether the key client complies with NIST SP800-131a. Specify on or off.

In the following example, the current directory contains the client keystore and the certificate chain. Enter the following command on one line:

```
/usr/lpp/mmfs/bin/mmgskkm trust --prefix DSM --pwd pwpkVormetric --out ksVormetric.keystore  
--label laccVormetric --fips off --nist on
```

The keystore of the key client contains the following items:

- Client credentials
- The certificate chain of the DSM key server as trusted certificates

Part 2: Configuring the DSM key server

The following instructions describe how to configure the DSM key server to communicate with an IBM Storage Scale key client.

In DSM, a *host* is a system to which DSM provides security services. In these instructions, the host is the IBM Storage Scale node that you are configuring for encryption. A DSM *domain* is an administrative group of one or more hosts. In these instructions, the domain contains the single IBM Storage Scale node. For more complex configurations, see the DSM product documentation.

1. Install a Key Management Interoperability Protocol (KMIP)-enabled license in DSM.

Important: You must complete this step before you create a DSM domain. For security reasons, you cannot create a KMIP-enabled domain in DSM until you install a KMIP-enabled license. For example, you cannot create a regular domain, install a KMIP-enabled license, and then convert the domain to a KMIP-enabled domain.

- a) On the DSM Management Console, click **System > License**.
- b) Select a KMIP-enabled license that you obtained from DSM.
- c) Click **Upload License File**.

The license is installed.

2. Create a DSM domain.

- a) On the DSM Management Console, click **Domains > Manage Domains**.
 - b) Follow the instructions to create a domain. Make sure that you configure the domain as **KMIP Supported**.
3. Create a Domain and Security Administrator for the new domain.
- Note:** In these instructions, a single Domain and Security Administrator is created who combines the responsibilities of administering the domain and controlling its security. For security reasons, you might want to create a Domain Administrator and a Security Administrator as separate roles. For more information, see the DSM documentation.
- a) Log in as the DSM System Administrator. On the Management Console, click **Administrators**.
 - b) On the **Administrators** page, click **Add**.
 - c) In the **Add Administrator** window, complete all the input fields except the **RSA User ID** field. In the **User Type** field, click **Domain and Security Administrator**.
- Note:** The passwords are temporary. The new administrator must enter a new password on the first login to the DSM Management Console.
- d) Click **OK**.
 - e) Limit the scope of the administrator's control to the domain that you created in Step 2.
4. Add the client CA certificates to the DSM trust store.
- If the client KMIP certificate is self-signed, this step is not required and you can skip to the Step 5, "Add a host to the domain".
- If the client KMIP certificate was signed by a CA, import the CA certificates that signed the client certificate to the DSM trust store. You must add all the CA certificates in the chain including the CA root certificate:
- a. Log in to DSM as Admin.
 - b. Navigate to **Settings > KMIP Trusted CA Certificates > Browse**.
 - c. Select the CA certificate file to import. The file can contain the entire certificate chain.
5. Add a host to the domain.
- a) Log in as the new administrator:
 - i) Enter a password when prompted.
 - ii) Select **I am a local domain administrator**.
 - iii) Enter or select the domain name from Step 2.
 - b) On the Management Console, click **Hosts > Hosts**.
 - c) On the **Hosts** screen, click **Add** to add a KMIP host. Set the **Host Name** to the name that you specified for the key client (the value for the cname parameter) when you created the client credentials in Part 1. In these instructions, the key client name is kcVormetric.
 - d) In the list of hosts, select the host that you created in the previous step. Click **Import KMIP Cert**. If no **Import KMIP Cert** button is displayed, verify that the DSM license is KMIP-enabled and that you created the domain after you installed the KMIP-enabled license.
 - e) In the window that opens, go through the directories of the IBM Storage Scale node to the directory that contains the client certificate file. Select the certificate file.
6. Create one or more keys for the client to use as master encryption keys (MEKs).
- The substeps in this step depend on the version of DSM that is installed:
- For DSM 5.2.3 or any version of DSM that is later than 5.2.3 and earlier than 6.0.2, follow these steps:
 - a. From the DSM Management Console page, click **Keys > Key Templates..** Follow the DSM instructions to create a key template. Select **AES256** as the key algorithm.
 - b. Create a key from the template. Specify a name for the key and then select the template.

- c. Make a note of the UUID of the key. You need it in Part 3.
- For DSM 6.2 and later, follow these steps:
 - a. On the DSM Management Console page, click **Keys > KMIP Objects**.
 - b. On the KMIP Objects page, click **Add** to create a new key.
 - c. On the Create KMIP Key window, enter a name for the key and select **AES_256** as the key type.
 - d. Click **OK** to close the window. The new key is added to the list of KMIP objects at the bottom of the KMIP Objects page.
 - e. Make a note of the UUID of the key. You need it in Part 3.

Part 3: Configuring the IBM Storage Scale node

1. Create an RKM.conf file and add a remote key management (RKM) stanza to it that contains the information that is necessary to communicate with the DSM key server.
 - a) On the IBM Storage Scale node, create a text file with the following path and name:

```
/var/mmfs/etc/RKM.conf
```

- b) Add a stanza with the following format:

```
stanzaName {
    type = KMIP
    kmipServerUri = tls://raclette.zurich.ibm.com:5696
    keyStore = /var/mmfs/etc/RKMcerts/ksVormetricDMS.p12
    passphrase = a_password
    clientCertLabel = a_label
}
```

where the rows of the stanza have the following meanings:

stanzaName

A name (RKM ID) for the stanza. Make a note of the name: you need it in the next step.

It is a good practice to use a format like the following one to ensure that the RKM ID is unique:

```
keyServerName_keyClientName
```

where *keyClientName* is the key client name from Part 1, Step 1. For example, the RKM ID for the key server and key client in these instructions is: *raclette_kcVormetric*.

type

Always KMIP for the DSM server.

kmipServerUri

The DNS name or IP address of the DSM server and the DSM SSL port. Multiple kmipServerUri entries may be added for high availability (HA), but note that the DSM servers must then be configured in an active-active setup. In the regular DSM HA setup, the passive failover nodes do not serve keys over KMIP. For more information, consult the DSM documentation.

keyStore

The path and name of the client keystore from Part 1.

passphrase

The password of the client keystore and client certificate from Part 1.

clientCertLabel

The label of the client certificate in the client keystore from Part 1.

2. Set up an encryption policy on the node that you are configuring for encryption.

- a) Create a policy that instructs GPFS to do the encryption tasks that you want.

The following policy is an example policy. It instructs IBM Storage Scale to encrypt all files in the file system with a file encryption key (FEK) and to wrap the FEK with a master encryption key (MEK):

```
RULE 'p1' SET POOL 'system' /* one placement rule is required at all times */
RULE 'Encrypt all files in file system with rule E1'
SET ENCRYPTION 'E1'
WHERE NAME LIKE '%'
RULE 'simpleEncRule' ENCRYPTION 'E1' IS
ALGO 'DEFAULTNISTSP800131A'
KEYS('01-10:raclette_kcVormetric')
```

In the last line, the character string within single quotation marks ('') is the key name. A *key name* is a compound of two parts in the following format:

KeyID:RkmID

where:

KeyID

Specifies the UUID of the master encryption key that you created in the DSM Management Console in Part 2.

RkmID

Specifies the name of the RKM stanza that you created in the /var/mmfs/etc/RKM.conf file in Step 1.

- b) Install the policy rule with the **mmchpolicy** command.



CAUTION: Installing a new policy with the **mmchpolicy** command removes all the statements in the previous policy. To add statements to an existing policy without deleting the previous contents, collect all policy statements for the file system into one file. Add the new statements to the file and install the contents of the file with the **mmchpolicy** command.

From now on, the encryption policy rule causes each newly created file to be encrypted with a file encryption key (FEK) that is wrapped in a master encryption key (MEK).

Part 4: Enabling encryption on other nodes

1. To replicate an encryption configuration on another node, you must copy some configuration files from the configured node to the target node:
 - a) Copy the /var/mmfs/etc/RKM.conf file to the same directory on the target node.
 - b) Copy the keystore files that the RKM file references to the same directories on the target node.
The recommended location for the keystore files on the configured node is /var/mmfs/etc/RKMcerts/.
2. To create a different encryption configuration on another node, follow the steps that are described in the preceding subtopics. Note the following design points:
 - On a single node:
 - The RKM.conf file can contain multiple stanzas. Each stanza represents a connection between a key client and a DSM host.
 - You can create multiple keystores.
 - Across different nodes:
 - The contents of RKM.conf files can be different.
 - The contents of keystores can be different.
 - If an encryption policy succeeds on one node and fails on another in the same cluster, verify that the failing node has the correct client keystore and stanza.

Remember: All nodes that mount a file system need to be able to access all the keys used in that file system.

Configuring encryption with the Thales CipherTrust Manager key server by using a local certificate authority

The topic describes a regular setup for encryption with Thales CipherTrust Manager by using a local certificate authority (CA).

Setting up an encryption environment with CipherTrust Manager as the key server requires IBM Storage Scale Data Management Edition 5.1 or later and a supported version of CipherTrust Manager. For more information, see the subtopic "Required software: Remote Key Management (RKM) server" in the help topic "[Preparation for encryption](#)" on page 742.

IBM Storage Scale supports CipherTrust Manager 2.5.x and 2.8 or later. CipherTrust Manager 2.6 and 2.7 are not supported. For more information, see [CipherTrust Manager Administration Guide](#).

Prerequisites:

The following requirements must be met on every IBM Storage Scale node that you configure for encryption:

- The node must have direct network access to the system where the key server is installed.
- The security-sensitive files that are created during the configuration must have the following characteristics:
 - They must be regular files that are owned by the root user.
 - They must be in the root group.
 - They must be readable and writable only by the user (mode '0600'). The following examples apply to the regular setup and the CipherTrust Manager setup:

```
-rw----- 1 root root 2446 Mar 20 12:15 /var/mmfs/etc/RKM.conf  
drw----- 2 root root 4096 Mar 20 13:47 /var/mmfs/etc/RKMcerts  
-rw----- 1 root root 3988 Mar 20 13:47 /var/mmfs/etc/RKMcerts/keystore_name.p12
```

The security-sensitive files include the following files:

- The RKM.conf file. For more information about this file, see "[The RKM.conf file and the RKM stanza](#)" on page 744.
- The files in the client keystore directory, which include the keystore file, the public and private key files for the client, and possibly other files. For more information about these files, see "[The client keystore directory and its files](#)" on page 746.



CAUTION:

- Ensure that the security-sensitive files are not lost or corrupted. IBM Storage Scale does not manage or replicate the files.
- Ensure that the passphrase for the client certificate file is not leaked through other means, such as the shell history.
- Client keystore files must be record-locked when the GPFS daemon starts. If the keystore files are stored on an NFS mount, the encryption initialization can hang. The cause is a bug that affects the way NFS handles record locking. If you encounter this problem, upgrade your version of NFS or store your keystore file on a local file system. If an upgrade is not possible and no local file system is available, use a RAM drive to store the keystore files.

See the following subtopics for instructions:

["Part 1 - Configuring the CipherTrust Manager key server"](#) on page 822

["Part 2 - Creating credentials for the key client"](#) on page 823

["Part 3 - Configuring the IBM Storage Scale node"](#) on page 824

["Part 4 - Enabling encryption on other nodes"](#) on page 825

["Part 5 - Configuring high-availability \(HA\) CipherTrust Manager cluster"](#) on page 826

Part 1 - Configuring the CipherTrust Manager key server

The following instructions describe how to configure the CipherTrust Manager key server to communicate with an IBM Storage Scale key client.

Install and configure a Key Management Interoperability Protocol (KMIP)-enabled CipherTrust Manager key server. For more information, see [CipherTrust Manager Administration Guide](#).

CipherTrust Manager supports the server certificate that is signed by:

- A local Certificate Authority (CA)
- An external Certificate Authority (CA)

Complete the following steps to configure the CipherTrust Manager server certificate:

1. Configuring the KMIP interface.

Interfaces are services that the CipherTrust Manager hosts. Since IBM Storage Scale uses the KMIP service, the server certificate on the KMIP interface must be configured.

- a) On the **CipherTrust Manager** window, click **Access Management > Registration Tokens** link. In the **Create New Registration Token** window, enter the token name and select the local Certificate Authority (CA).
- b) Click **Create New Registration Token** and click **Copy** to copy the generated registration token, click **Add Token**.
- c) To edit and configure the KMIP interface, complete the following steps:
 - i) Open the **CipherTrust Manager** window, click **Admin Settings > Interfaces**.
 - ii) Expand the KMIP interface by clicking the ellipsis (...), then **View/Edit**.
 - iii) In the **Configure KMIP** page, select **Auto Registration** and paste the registration token.
 - iv) Select the mode as **TLS, verify client cert, username taken from client cert, auth request is optional**.
 - v) Ensure that **Username Location in Certificate** is set to **CN**.
 - vi) Under **Local Trusted CAs**, ensure that the local CipherTrust Manager CA is selected, click **Update**.

2. Creating a key client certificate.

A key client can request master encryption keys from the key server. The key client certificate must be signed by the same CA that signed the server certificate, in this case the CipherTrust Manager local CA. To generate a client certificate in CipherTrust Manager, complete the following steps:

- a) On the **CipherTrust Manager** window, click **CA > Local**. Click the name of the Local Certificate Authority to view its details.
- b) Click **Issue Certificate**.
- c) Enter the **Display Name, Common Name**, and other client certificate information. Make a note of the client's common name (CN), it is needed in the further steps. Save the generated certificate sign request (CSR) and the private key.
- d) Click **Issue Certificate**. The newly generated client certificate appears in the list of certificates that are generated by the local CA.
- e) Expand the client certificate by clicking the ellipsis (...), then click **Download**, save the client certificate. The client certificate is needed in [Part 3](#).
- f) Download the CipherTrust Manager local CA. On the **CipherTrust Manager** window, click **CA > Local**.
- g) Expand the local CA that is being used by clicking the ellipsis (...), then click **Download**, and then save the CA certificate chain. The certificate chain is needed in [Part 3](#).

3. Adding the key client as a user in CipherTrust Manager.

A user is an authenticated entity or a server where IBM Storage Scale is installed. The user can make KMIP calls to CipherTrust Manager to retrieve encryption keys. To create a user, complete the following steps:

- a) In the **CipherTrust Manager** window, click **Access Management > Users > Create New User**. The Username must be similar to the client's common name (CN) specified when creating the client's certificate.
 - b) Click the newly created user and expand **GROUPS**. Search for **Key Users** and **Key Admins** groups and add the user.
4. Creating and updating an encryption key.

IBM Storage Scale 5.1.4 or later supports maximum 65 characters for the Universally Unique Identifier (UUID) of an encryption key. By default, CipherTrust Manager creates keys with 65 characters UUIDs. Complete the following steps to create a key with 65 characters UUID.

- a) In the **CipherTrust Manager** window, click **Keys > Create New Key**.
- b) Enter the key name and other information and client on the **Create** window.
- c) The key ID is displayed. Copy the key ID to use in the IBM Storage Scale encryption policy in [Part 3](#).

IBM Storage Scale 5.1.3 or earlier supports maximum 60 characters for the Universally Unique Identifier (UUID) of an encryption key. Complete the following steps to create a key with 60 characters UUID:

- a) On the **CipherTrust Manager** window, click **API** to open the API Playground.
- b) Click **Authenticate**. Specify the admin username and password, then click **Post**. The session is valid for 300 seconds.
- c) On the sidebar, search for **Keys**, then click **Create - Post**.
- d) In the body section, specify a key UUID length of 60 characters, for example:

```
{  
  "name": "ScaleKey"  
  "idSize" : 60  
}
```

- e) Click **POST** to create the key.
 - f) The key ID is displayed. Copy the key ID to use in the IBM Storage Scale encryption policy in [Part 3](#).
- Updating the key attributes.
- a) On the **CipherTrust Manager** window, select **Keys** and click the newly created key.
 - b) Ensure that **Exportable** option is checked.
 - c) Expand **KEY ACCESS** and add the newly created user as the **Key Owner**.

Part 2 - Creating credentials for the key client

To create credentials for the key client, complete to the following steps:

1. Copy the CA chain, the client certificate, and private key files generated in [Part 1, Step 1](#) from CipherTrust Manager to the IBM Storage Scale node.
2. CipherTrust Manager generates the client private key in the Elliptic Curve (EC) format. IBM Storage Scale 5.1.5 or later supports an EC private key to generate the client keystore. When running IBM Storage Scale 5.1.4 or earlier, convert the EC private key format into the PKCS8 format, for example:

```
openssl pkcs8 -topk8 -nocrypt -in EC_privFile -out privFile
```

3. Create a PKCS#12 keystore to store the certificate and private key of the client in it. Issue the following command in a single line:

```
mmgskkm store --cert scaleclient.cert --priv scaleclient.priv --chain CA_chain.pem --label client --pwd clientpassword --out ctmclient.p12
```

where:

--cert certFile
 Specifies the client certificate file that is created in [Part 1, Step 2](#).

--priv privFile
 Specifies the private key file that you created in [Part 1, Step 2](#).

--label label
 Specifies the label under which the private key is stored in the keystore. Use a common name that was used when you created the client in CipherTrust Manager.

--pwd pwd
 Specifies the password of the keystore. You can use the same password that you specified for the private key in [Part 1, Step 2](#).

--out keystore
 The file name of the keystore.

In the following example, the current directory contains the client credentials and the CA chain from [Part 1, Step 2](#):

```
mmgskkm store -cert scaleclient.cert --priv scaleclient.priv --chain CA_chain.pem --
label client --pwd clientpassword --out ctmclient.p12
```

The output file is a keystore that contains the client credentials of the key client.

Important: The keystore must be record-locked when the GPFS daemon starts. If the keystore files are stored on an NFS mount, the encryption initialization can hang. The cause is a bug that affects the way NFS handles record locking. If you encounter this problem, upgrade your version of NFS or store your keystore file on a local file system. If an upgrade is not possible and no local file system is available, use a RAM drive to store the keystore files.

Part 3 - Configuring the IBM Storage Scale node

1. Create an RKM.conf file and add a remote key management (RKM) stanza to it. The stanza contains the necessary information to communicate with the CipherTrust Manager key server.
 - a) On the IBM Storage Scale node, create a text file with the following path and name:

```
/var/mmfs/etc/RKM.conf
```

- b) Add a stanza with the following format:

```
stanzaName {
    type = KMIP
    kmipServerUri = tls://ctmServer1.ibm.com:5696
    keyStore = /var/mmfs/etc/RKMcerts/ctmClient.p12
    passphrase = a_password
    clientCertLabel = a_label
}
```

where the rows of the stanza have the following meanings:

stanzaName

A name (RKM ID) for the stanza. Make a note of the name, you need it in the next step. It is a good practice to use a format like the following one to ensure that the RKM ID is unique:

```
keyServerName_keyClientName
```

where *keyClientName* is the key client name from [Part 1, Step 1](#). For example, the RKM ID for the key server and key client in these instructions is: *ctmServer1_ctmClient*.

type

Always KMIP for the CipherTrust Manager server.

kmipServerUri

The DNS name or IP address of the CipherTrust Manager and the CipherTrust Manager SSL port. Multiple kmipServerUri entries can be added for high-availability (HA).

keyStore

The path and name of the client keystore from [Part 1](#).

passphrase

The password of the client keystore and client certificate from [Part 1](#).

clientCertLabel

The label of the client certificate in the client keystore from [Part 1](#).

2. Set up an encryption policy on the node that you are configuring for encryption.

- a) Create a policy that instructs GPFS to do the encryption tasks that you want.

The following policy is an example of a policy. It instructs IBM Storage Scale to encrypt all files in the file system with a file encryption key (FEK) and to wrap the FEK with a master encryption key (MEK):

```
RULE 'p1' SET POOL 'system'  
RULE 'Encrypt all files in file system with rule E1'  
SET ENCRYPTION 'E1'  
WHERE NAME LIKE '%'  
RULE 'simpleEncRule' ENCRYPTION 'E1' IS  
ALGO 'DEFAULTNISTSP800131AFAST'  
KEYS(' 5c62fa7fb9e2e5670f4fb63b18ee2ab73b8c5ea9a5ff206338d2f8025ce9:ctmServer1_ctmClient)
```

In the last line, the character string within single quotation marks ('') is the key name. A *key name* is a compound of two parts in the following format:

KeyID:RkmID

where:

KeyID

Specifies the UUID of the master encryption key that you created in **CipherTrust Manager** window or the API Playgroun in [Part 1](#).

RkmID

Specifies the name of the RKM stanza that you created in the /var/mmfs/etc/RKM.conf file in [Step 1](#).

- b) Install the policy rule by issuing the **mmchpolicy** command.



CAUTION: Installing a new policy with the mmchpolicy command removes all the statements in the previous policy. To add statements to an existing policy without deleting the previous contents, collect all policy statements for the file system into one file by using the **mmlspolicy** command. Add the new statements to the file and install the contents of the file with the mmchpolicy command.

Encryption policy rule causes each newly created file to be encrypted with a file encryption key (FEK) that is wrapped in a master encryption key (MEK) that corresponds to the matching encryption rule.

Part 4 - Enabling encryption on other nodes

1. To replicate an encryption configuration on another node, you must copy some configuration files from the configured node to the target node:
 - a) Copy the /var/mmfs/etc/RKM.conf file to the same directory on the target node.
 - b) Copy the keystore files that the RKM file references to the same directories on the target node.
The recommended location for the keystore files on the configured node is /var/mmfs/etc/RKMcerts/.
2. To create a different encryption configuration on another node, follow the steps that are described in the preceding subtopics. Note the following design points:
 - On a single node:
 - The RKM.conf file can contain multiple stanzas. Each stanza represents a connection between a key client and a CipherTrust Manager server.

- You can create multiple keystores.
- Across different nodes:
 - The contents of RKM.conf files can be different.
 - The contents of keystores can be different.
 - If encrypted files are accessible on one node and inaccessible on another in the same cluster, verify that the failing node has the correct client keystore and stanza.

Remember: All nodes that mount a file system need to be able to access all the keys used in that file system.

Part 5 - Configuring high-availability (HA) CipherTrust Manager cluster

1. Install multiple CipherTrust Manager servers. To create a high-availability cluster, perform the following steps on the primary CipherTrust Manager server:
 - a) Open **CipherTrust Manager** window, select **Admin Settings > Cluster**.
 - b) Click **Manage cluster** and select **Add cluster**.
 - c) Enter the primary server's private and/or public IP addresses and click **Save**.
 - d) To add backup servers to the cluster, click **Manage cluster > Add node** and specify the backup server information.
 - e) Follow the prompt to log in to the backup server and join the cluster.
 - f) Repeat steps d and e for each backup server.

For more information, see [CipherTrust Manager Administration Guide](#).
2. Update the RKM.conf file to list the backup CipherTrust Manager servers under the keyword kmipServerUri2,kmipServerUri3, as shown in the following example:

```
ctmServer1_ctmClient {
  type = KMIP
  kmipServerUri = tls://ctmServer1.ibm.com:5696
  kmipServerUri2 = tls://ctmServer2.ibm.com:5696
  kmipServerUri3 = tls://ctmServer3.ibm.com:5696
  keyStore = /var/mmfs/etc/RKMcerts/ctmClient.p12
  passphrase = a_password
  clientCertLabel = a_label
}
```

Configuring encryption with the Thales CipherTrust Manager key server by using an external certificate authority

The topic describes a regular setup for encryption with Thales CipherTrust Manager by using an external certificate authority (CA).

Setting up an encryption environment with CipherTrust Manager as a key server requires IBM Storage Scale Data Management Edition 5.1 or later and a supported version of CipherTrust Manager. For more information, see the subtopic "Required software: Remote Key Management (RKM) server" in the help topic ["Preparation for encryption" on page 742](#).

IBM Storage Scale supports CipherTrust Manager v 2.5.x and v 2.8 or later. CipherTrust Manager v 2.6 and v2.7 are not supported. For more information, see [CipherTrust Manager Administration Guide](#).

Prerequisites:

The following requirements must be met on every IBM Storage Scale node that you configure for encryption:

- The node must have direct network access to the system where the key server is installed.
- The security-sensitive files that are created during the configuration must have the following characteristics:
 - They must be regular files that are owned by the root user.

- They must be in the root group.
- They must be readable and writable only by the user (mode '0600'). The following examples apply to the regular setup and the CipherTrust Manager setup:

```
-rw----- 1 root root 2446 Mar 20 12:15 /var/mmfs/etc/RKM.conf
drw----- 2 root root 4096 Mar 20 13:47 /var/mmfs/etc/RKMcerts
-rw----- 1 root root 3988 Mar 20 13:47 /var/mmfs/etc/RKMcerts/keystore_name.p12
```

The following files are security-sensitive files:

- The RKM.conf file. For more information, see [“The RKM.conf file and the RKM stanza” on page 744](#).
- The files in the client keystore directory, which include the keystore file, the public and private key files for the client, and possibly other files. For more information, see [“The client keystore directory and its files” on page 746](#).



CAUTION:

- Ensure that the security-sensitive files are not lost or corrupted. IBM Storage Scale does not manage or replicate the files.
- Ensure that the passphrase for the client certificate file is not leaked through other means, such as the shell history.
- Client keystore files must be record-locked when the GPFS daemon starts. If the keystore files are stored on an NFS mount, the encryption initialization can hang. The cause is a bug that affects the way NFS handles record locking. If you encounter this problem, upgrade your version of NFS or store your keystore file on a local file system. If an upgrade is not possible and no local file system is available, use a RAM drive to store the keystore files.

See the following subtopics for instructions:

[“Part 1 - Configuring the CipherTrust Manager key server” on page 827](#)

[“Part 2 - Creating credentials for the key client” on page 829](#)

[“Part 3 - Configuring the IBM Storage Scale node” on page 830](#)

[“Part 4 - Enabling encryption on other nodes” on page 831](#)

[“Part 5 - Configuring a high-availability \(HA\) CipherTrust Manager cluster” on page 831](#)

Part 1 - Configuring the CipherTrust Manager key server

The following instructions describe how to configure the CipherTrust Manager key server to communicate with an IBM Storage Scale key client.

Install and configure a Key Management Interoperability Protocol (KMIP)-enabled CipherTrust Manager key server. For more information, see [CipherTrust Manager Administration Guide](#).

CipherTrust Manager supports the server certificate that is signed by:

- A local Certificate Authority (CA)
- An external Certificate Authority (CA)

Complete the following steps to configure CipherTrust Manager server certificate:

1. Adding an external certificate authority (CA).

a) On the **CipherTrust Manager** window, select **CA > External**.

b) Click **Add External CA** and paste the contents of the CA certificate chain that is provided by the external certificate authority.

2. Configuring the KMIP interface.

Interfaces are services that the CipherTrust Manager hosts. Since IBM Storage Scale uses the KMIP service, the server certificate on the KMIP interface must be configured.

To configure the KMIP interface, enable automatic registration by creating a registration token as follows:

- a) On the **CipherTrust Manager** window, click **Access Management > Registration Tokens**. In the **Create New Registration Token** window, enter the token name, and select the local Certificate Authority (CA).
- b) Click **Create New Registration Token** and click **Copy** to copy the generated registration token. Click **Add Token**.

To edit and configure the KMIP interface, complete the following steps:

- a) Open the **CipherTrust Manager** window, click **Admin Settings > Interfaces**.
- b) Expand the KMIP interface by clicking the ellipsis (...), then click **View/Edit**.
- c) On the **Configure KMIP** page, select **Auto Registration** and paste the registration token.
- d) Select the mode as **TLS, verify client cert, username taken from client cert, auth request is optional**.
- e) Ensure that **Username Location in Certificate** is set to **CN**.
- f) Under **Local CA for Automatic Server Certificate Generation**, select **Turn off auto generation from a Local CA**. If this option is not turned off, the CA signed server certificate can be replaced by a server certificate that is signed by the local CA after a restart.
- g) Under **External Trusted CAs**, select the external CA, click **Add**, and then **Update**.
- h) Expand the KMIP Interface by clicking the ellipsis, then click **Upload New Certificate**. Upload the server certificate and private key files or paste the files into the text editor in PEM or base64 encoded PKCS12 format.

3. Creating a key client certificate.

A key client can request master encryption keys from the key server. The key client certificate must be signed by the same external certificate authority (CA) that signed the server certificate. Create the client CA-signed certificate and the client-private key. The generated client certificate and private key are used in [Part 2](#).

4. Adding the key client as a User in CipherTrust Manager.

A user is an authenticated entity or a server where IBM Storage Scale is installed. The user can make KMIP calls to CipherTrust Manager to retrieve encryption keys. To create a user, complete the following steps:

- a) In the **CipherTrust Manager** window, click **Access Management > Users > Create New User**. The Username must be similar to the client's common name (CN) specified when you created the client's certificate.
- b) Click the newly created user and expand **GROUPS**. Search for **Key Users** and **Key Admins** groups and add the user.

5. Creating and updating an encryption key.

IBM Storage Scale 5.1.4 or later supports maximum 65 characters for the Universally Unique Identifier (UUID) of an encryption key. By default, CipherTrust Manager creates keys with 65 characters UUIDs. To create a key with 65 characters UUID, complete the following steps:

- a) In the **CipherTrust Manager** window, click **Keys > Add Key**.
- b) Enter the key name and other information and client on **Add Key** window.
- c) The key ID is displayed. Copy the key ID to use in the IBM Storage Scale encryption policy in [Part 3](#).

IBM Storage Scale 5.1.3 or earlier supports maximum 60 characters for the Universally Unique Identifier (UUID) of an encryption key. To create a key with 60 characters UUID, complete the following steps:

- a) On the **CipherTrust Manager** window, click **API** to open the API Playground.
- b) Click **Authenticate**. Specify the admin username and password, then click **Post**. The session is valid for 300 seconds.
- c) On the sidebar, search for **Keys**, then click **Create - Post**.
- d) In the body section, specify a key UUID length of 60 characters, for example:

```
{  
  "name": "ScaleKey"  
  "idSize" : 60  
}
```

- e) Click **POST** to create the key.
- f) The key ID is displayed on the UI. Copy the key ID to use in the IBM Storage Scale encryption policy in [Part 3](#).

Updating the key attributes.

- a) On the **CipherTrust Manager** window, navigate to **Keys** and click the newly created key.
- b) Ensure that **Exportable** option is checked.
- c) Expand **KEY ACCESS** and add the newly created user as the **Key Owner**.

Part 2 - Creating credentials for the key client

To create credentials for the key client, complete the following steps:

1. Copy the CA certificate chain, the CA-signed client certificate, and the client key to the IBM Storage Scale node.
2. CipherTrust Manager generates the client-private key in the Elliptic Curve (EC) format. IBM Storage Scale 5.1.5 or later supports an EC private key to generate the client-private keystore. When running IBM Storage Scale or earlier, convert the EC private key format into PKCS8 format, for example:

```
openssl pkcs8 -topk8 -nocrypt -in EC_privFile -out privFile
```

3. Create a PKCS#12 keystore to store the certificate and private key of the client in it. Issue the following command in a single line:

```
/usr/lpp/mmf/bin/mmgskkm store --cert certFile --priv privFile --chain CA_chain.pem --label label --pwd pwd --out keystore
```

Where:

--cert certFile

Specifies the client certificate file that is created in [Part 1, Step 2](#).

--priv privFile

Specifies the private key file that you created in [Part 1, Step 2](#).

--label label

Specifies the label under which the private key is stored in the keystore. Use a common name that was used when you created the client in CipherTrust Manager.

--pwd pwd

Specifies the password of the keystore. You can use the same password that you specified for the private key in [Part 1, Step 2](#).

--out keystore

The file name of the keystore.

In the following example, the current directory contains the client credentials and the CA chain from [Part 1, Step 2](#):

```
mmgskkm store --cert scaleclient.cert --priv scaleclient.priv --chain CA_chain.pem --label client --pwd clientpassword --out ctmcclient.p12
```

The output file is a keystore that contains the client credentials of the key client.

Important: The keystore must be record-locked when the GPFS daemon starts. If the keystore files are stored on an NFS mount, the encryption initialization can hang. The cause is a bug that affects the way NFS handles record locking. If you encounter this problem, upgrade your version of NFS or store your keystore file on a local file system. If an upgrade is not possible and no local file system is available, use a RAM drive to store the keystore files.

Part 3 - Configuring the IBM Storage Scale node

To configure IBM Storage Scale node, complete the following steps:

1. Create an RKM.conf file and add a remote key management (RKM) stanza to it. The stanza contains the necessary information to communicate with the CipherTrust Manager key server.
 - a) On the IBM Storage Scale node, create a text file with the following path and name:

```
/var/mmfs/etc/RKM.conf
```

- b) Add a stanza with the following format:

```
stanzaName {
    type = KMIP
    kmipServerUri = tls://ctmServer1.ibm.com:5696
    keyStore = /var/mmfs/etc/RKMcerts/ctmClient.p12
    passphrase = a_password
    clientCertLabel = a_label
}
```

where, the rows of the stanza have the following meanings:

stanzaName

A name (RKM ID) for the stanza. Make a note of the name: you need it in the next step.

It is a good practice to use a format like the following one to ensure that the RKM ID is unique:

```
keyServerName_keyClientName
```

where *keyClientName* is the key client name from [Part 1, Step 1](#). For example, the RKM ID for the key server and key client in these instructions is: `ctmServer1_ctmClient`.

type

Always KMIP for the CipherTrust Manager server.

kmipServerUri

The DNS name or IP address of the CipherTrust Manager and the CipherTrust Manager SSL port. Multiple kmipServerUri entries can be added for high-availability (HA).

keyStore

The path and name of the client keystore from [Part 1](#).

passphrase

The password of the client keystore and client certificate from [Part 1](#).

clientCertLabel

The label of the client certificate in the client keystore from [Part 1](#).

2. Set up an encryption policy on the node that you are configuring for encryption.

- a) Create a policy that instructs GPFS to do the encryption tasks that you want.

The following policy is an example of a policy. It instructs IBM Storage Scale to encrypt all files in the file system with a file encryption key (FEK) and to wrap the FEK with a master encryption key (MEK):

```
RULE 'p1' SET POOL 'system'
RULE 'Encrypt all files in file system with rule E1'
SET ENCRYPTION 'E1'
WHERE NAME LIKE '%'
RULE 'simpleEncRule' ENCRYPTION 'E1' IS
ALGO 'DEFAULTNISTSP800131AFAST'
KEYS(' 5c62fa7fb9e2e5670f4fb63b8c5ea9a5ff206338d2f8025ce9:ctmServer1_ctmClient')
```

In the last line, the character string within single quotation marks (') is the key name. A *key name* is a compound of two parts in the following format:

```
KeyID:RkmID
```

where:

KeyID

Specifies the UUID of the master encryption key that you created in **CipherTrust Manager** window or the API Playgroud in [Part 1](#).

RkmID

Specifies the name of the RKM stanza that you created in the /var/mmfs/etc/RKM.conf file in [Step 1](#).

- b) Install the policy rule by issuing the **mmchpolicy** command.



CAUTION: Installing a new policy with the **mmchpolicy** command removes all the statements in the previous policy. To add statements to an existing policy without deleting the previous contents, collect all policy statements for the file system into one file by using the **mmlspolicy** command. Add the new statements to the file and install the contents of the file with the **mmchpolicy** command.

The encryption policy rule causes each newly created file to be encrypted with a file encryption key (FEK) that is wrapped in a master encryption key (MEK) that corresponds to the matching encryption rule.

Part 4 - Enabling encryption on other nodes

1. To replicate an encryption configuration on another node, you must copy some configuration files from the configured node to the target node:
 - a) Copy the /var/mmfs/etc/RKM.conf file to the same directory on the target node.
 - b) Copy the keystore files that the RKM file references to the same directories on the target node.
The recommended location for the keystore files on the configured node is /var/mmfs/etc/RKMcerts/.
2. To create a different encryption configuration on another node, follow the steps that are described in the preceding subtopics. Note the following design points:
 - On a single node:
 - The RKM.conf file can contain multiple stanzas. Each stanza represents a connection between a key client and a CipherTrust Manager server.
 - You can create multiple keystores.
 - Across different nodes:
 - The contents of RKM.conf files can be different.
 - The contents of keystores can be different.
 - If encrypted files are accessible on one node and inaccessible on another in the same cluster, verify that the failing node has the correct client keystore and stanza.

Remember: All nodes that mount a file system need to be able to access all the keys used in that file system.

Part 5 - Configuring a high-availability (HA) CipherTrust Manager cluster

1. Install multiple CipherTrust Manager servers. To create a high-availability cluster, perform the following steps on the primary CipherTrust Manager server:
 - a) Open **CipherTrust Manager** window, select **Admin Settings > Cluster**.
 - b) Click **Manage Cluster** and select **Add cluster**.
 - c) Enter the primary server's private and/or public IP addresses then click **Add cluster**.
 - d) To add backup servers to the cluster, click **Manage cluster > Add node** and specify the backup server information.
 - e) Follow the prompt to log in to the backup server and join the cluster.

f) CipherTrust Manager HA setup process does not import the server certificate into the backup servers. When the server certificate is signed by an external CA, import the server certificate and the server-private key into the KMIP interface on the backup server.

g) Repeat steps d through f for each backup server.

For more information, see [CipherTrust Manager Administration Guide](#).

2. Update the RKM.conf file to list the backup CipherTrust Manager servers under the keyword kmipServerUri2,kmipServerUri3, as shown in the following example:

```
ctmServer1_ctmClient {  
    type = KMIP  
    kmipServerUri = tls://ctmServer1.ibm.com:5696  
    kmipServerUri2 = tls://ctmServer2.ibm.com:5696  
    kmipServerUri3 = tls://ctmServer3.ibm.com:5696  
    keyStore = /var/mmfs/etc/RKMcerts/ctmClient.p12  
    passphrase = a_password  
    clientCertLabel = a_label  
}
```

Certificate expiration warnings

IBM Storage Scale writes warning messages into the mmfs.log file for digital certificates that are nearing their expiration dates.

Warnings are issued for both RKM server certificates and key client certificates.

Note: To renew an expired server or client certificate, see the topic [Renewing client and server certificates](#).

Warnings for an RKM server certificate

A warning message for an RKM server certificate that is approaching its expiration date contains the date and time of expiration and the IP address and port of the RKM server, as in the following example. In the log file this message would be printed all on one line:

```
2018-08-01_11:45:09.341-0400: GPFS: 6027-3732 [W] The server certificate for key  
server 192.168.9.135 (port 5696) will expire at Aug 01 12:03:32 2018 EDT (-0400).
```

With this information you can log on to the specified RKM server and find the server certificate that is approaching expiration.

Warnings for a key client certificate

The warning message for a key client certificate that is approaching its expiration date contains the date and time of the expiration, the IP address and port of the RKM server to which the key client has a connection, the label of the client certificate, and the RKM ID. In the log file this message would be printed all on one line:

```
2020-11-04_13:55:07.838-0400: [W] The client certificate with label 'client1' for key server  
with RKM ID 'RKM1' (192.168.9.135:5696) will expire at Nov 04 16:39:59 2020 EDT (-0400).
```

The procedure for identifying an expiring client certificate based on the RKM server information in the error message depends on two circumstances:

- Whether more than one key client in the cluster has a connection with the RKM server that is specified in the error message.
- Whether the encryption environment of the cluster is configured by the simplified setup method or the regular setup method.

The following instructions assume that only one key client in the cluster has a connection with the specified RKM server:

- **Simplified method:** If the encryption environment is configured by the simplified method, follow these steps:

1. Make a note of the following information:
 - The expiration date of the client certificate from the warning message.
 - The IP address and port of the RKM server from the error message.
 - The host name of the RKM server that uses that IP address and port. Look this item up in your system information.
2. On the command line of a node in the cluster, issue the following command to list the key clients for the RKM server:

```
mmkeyserv client show -server <host_ID>
```

where *<host_ID>* is the IP address or host name of the RKM server from Step 1.

3. For each key client the command displays a block of information that includes the client certificate label, the host name or IP address and the port of the RKM server, and other information.
4. This set of instructions assumes that only one key client in the cluster has a connection with the specified RKM server. Therefore, in Step 3 the command displays only one block of information. The label that is listed in this block of information is the label of the client certificate that is approaching expiration.

- **Regular method:** If the encryption environment is configured by the regular method, follow these steps:

1. Make a note of the following information:
 - The expiration date of the client certificate from the warning message
 - The IP address and port of the RKM server from the error message.
 - The host name of the RKM server that uses that IP address and port. Look this item up in your system information.
2. On a node of the cluster that accesses encrypted files – that is, on a node that is successfully configured for encryption – open the RKM.conf file with a text editor. For more information about the RKM.conf file, see the topic “[Preparation for encryption](#)” on page 742.
3. In the RKM.conf file, follow these steps:
 - a. Find the stanza that contains the host name or IP address and the port of the RKM server from Step 1. This information is specified in the **kmpServerURI** parameter of the stanza.
 - b. The client certificate label that is specified in that same stanza is the label of the client certificate that is approaching expiration.
 - c. Make a note of the path of the keystore and the keystore password that are also specified in the stanza. You can use this information to open the keystore with a tool such as the openssl key-management utility and inspect the certificate.

If more than one key client in the cluster might have a connection with the RKM server that is specified in the error message, then you must identify each such key client and search its keystore to find the certificate that is approaching expiration. The following instructions are for both the simplified setup method and the regular setup method:

1. Make a note of the expiration date of the client certificate and the IP address and port of the RKM server in the error message. Also look up the host name of the RKM server.
2. List the stanzas of the RKM.conf file:

- For the simplified setup method, issue the following command from the command line:

```
mmkeyserv rkm show
```

- For the regular setup method, open the RKM.conf file with a text editor. You must do this step on a node that is configured for encryption. For more information about the RKM.conf file, see the topic “[Preparation for encryption](#)” on page 742.

3. Find the stanza or stanzas that contain the host name or IP address of the RKM server from Step 1. For each such stanza, make a note of the client certificate label, the path of the keystore file, and the password to the keystore file.
4. Open each keystore file from Step 3 with a tool such as the openssl key-management utility. In the keystore file, find the client certificate label or labels from Step 3 and verify whether each client certificate is approaching expiration.

To renew an expired client certificate, see the topic [“Renewing client and server certificates” on page 835](#).

Only certificates that are in use are checked

IBM Storage Scale checks certificate expiration dates only when the certificates are being used to authenticate a connection between a key client and a key server.

IBM Storage Scale checks the certificate expiration dates of a key client and its RKM server at regular intervals, currently every 15 minutes. The first check occurs when the key client connects with the server to obtain a master encryption key (MEK), which it stores in a local cache on the network node. Subsequent checks occur regularly as the key client periodically reconnects with the RKM server so that it can refresh the MEK in the local cache. The current refresh interval is 15 minutes.

IBM Storage Scale does not check the certificate expiration dates of client or server certificates that are not currently being used in this way. This category includes not-in-use client certificates in local keystores and not-in-use server certificates for RKM backup servers.

Frequency of warnings

The frequency of warnings increases as the expiration date nears, as the following table illustrates:

<i>Table 65. Frequency of warnings</i>	
Time before expiration	Frequency of warnings
More than 90 days	No warnings are logged.
30 - 90 days	Every seven days.
7 days - 30 days	Every 24 hours.
24 hours - 7 days	Every 60 minutes.
Less than 24 hours	Every 15 minutes.

A first warning is issued when both of the following conditions become true:

- At least 75 percent of the certificate validity period has passed.
- The time that remains falls within one of the warning windows.

Subsequent warnings are issued with the frequency that is listed in the second column of the preceding table. For example, if the validity period is 30 days and begins at midnight on March 1, then the warnings are issued as shown in the following list:

First warning: March 22 at 12:00 noon (.75 * 30 days = 22.5 days).

Second warning: March 23 at 12:00 noon (7.5 days remaining).

Third warning: March 24 at 12:00 noon (6.5 days remaining).

Warnings: Every 60 minutes from March 24 at 1:00 PM until March 29 at 12:00 midnight.

Warnings: Every 15 minutes from March 29 at 12:15 AM until March 30 at midnight.

Limitations

This feature has the following restrictions and limitations:

- Warnings are logged only on nodes that access encrypted files.

- Warnings are logged only for certificates that are used to authenticate a connection between a key client and an RKM server that is still active.
- Warning messages identify only the type of certificate (client or server) and the IP address and port of the RKM server.

Renewing client and server certificates

Learn how to renew IBM Storage Scale client and server certificates.

During encryption, the GPFS daemon acts as a key client and requests master encryption keys (MEKs) from a Remote Key Management (RKM) server. The supported RKM servers are IBM Security Key Lifecycle Manager (SKLM) and Thales Vormetric Data Security Manager (DSM).

When a digital client or server certificate expires, the IBM Storage Scale client cannot access encrypted files, because it can no longer retrieve MEKs from the RKM server. The following topics describe how to recognize certificate expiration errors and how to renew client and server certificates.

MEKs do not expire unless they are explicitly removed from a key server.

The following table shows the default lifetimes of client and server certificates:

Table 66. Comparing default lifetimes of key server and key client certificates		
Item	Type of certificate	Default lifetime
IBM Storage Scale	Client	3 years ¹
IBM Security Key Lifecycle Manager (SKLM)	Server	3 years
Thales Vormetric Data Security Manager (DSM)	Server	10 years

¹You can create an IBM Storage Scale client certificate with a shorter or longer lifetime by issuing the **mmkeyserv client create** command with the **--days** option.

Certificate expiration dates and error messages

Learn how to check the expiration dates of Remote Key Management (RKM) server certificates and key client certificates. Also, learn to recognize the error messages that report that an RKM server certificate or a key client certificate expired.

Checking the expiration dates of RKM server and key client certificates

- If you are using the simplified setup method in IBM Storage Scale 5.0.3 or later, follow these steps for the RKM server certificate and the key client certificate:

RKM server certificate

Issue the following command:

```
mmkeyserv server show ServerName
```

where, *ServerName* is the host name or IP address that you specified for the server in the **mmkeyserv server add** command. In the following example, the server name is hs21n62. The expiration date of the server is displayed in the line that begins **KMIP Certificate Expiration**:

```
# mmkeyserv server show hs21n62
hs21n62.gpfs.net
Type: ISKLM
IPA: 192.168.38.14
User ID: SKLMAAdmin
REST_port: 9443
Label: 2_hs21n62
NIST: on
FIPS1402: on
```

```

Backup Key Servers:
Distribute: yes
Retrieval Timeout: 60
Retrieval Retry: 3
Retrieval Interval: 10000
REST Certificate Expiration: 2035-02-01 21:35:02 (-0500)
KMIP Certificate Expiration: 2028-04-24 22:51:31 (-0400)

```

Key client certificate

Issue the following command:

```
mmkeyserv client show ClientName
```

where, *ClientName* is the name that you specified for the client in the `mmkeyserv client create` command. In the following example, the client name is `sklm4Client`. The expiration date of the client is displayed in the line that begins `Certificate Expiration`:

```

# mmkeyserv client show sklm4Client
sklm4Client
Label: sklm4Client
Key Server: hs21n62.gpfs.net
Tenants: Newsklm4Tenant,sklm4Tenant
Certificate Expiration: 2023-03-27 10:45:10 (-0400)

```

- If you are using the regular setup method, follow these steps for the RKM server certificate and the key client certificate:

RKM server certificate

If the RKM server is running with a self-signed certificate, follow these steps:

1. Issue the `mmsklmconfig restcert` command to retrieve the server certificate. For more information, see “[Renewing expired client certificates](#)” on page 845.
2. Issue the `mmgskkm print` command to display the contents of the server certificate. For more information, see “[Renewing expired client certificates](#)” on page 845.
3. In the `mmgskkm print` command output, find the expiration date for the server certificate. In the following example, the expiration date is on the final line, which begins “Valid until”:

```

# mmsklmconfig restcert --host hs21n62 --port 5696 --prefix sklmCert --keystore
sklm4Client.p12 --keypass keystorePass
# ls -ltr sklmCert0.cert
-rw-r--r--. 1 root root 1017 Jul 27 00:55 sklmCert0.cert
# mmgskkm print --cert sklmCert0.cert
Serial number: 2f2409efce9447
SHA-256 digest:
0c9fabf65ab3bea6259af4829cf4027db1395d46a71d49631af7c2a3454ff20d
Signature:
788c8c9a3ec673ac7276283f6720ff4c910f9235042f2959eb37a466277d11a9f085112e28126b05c64516
50c9595bd21ab48aabac1ac1fab4a8e945f3df2de12c82f57c44e13d983305c3a7ba41d8d565c9db6a545
981c16b12af7538f85740e6d0500266cec9fc2cf4b878c7ef12d18fd10e43c0933d246ab825dc5f059c6bb
0e82f5fabd302e661584deb63b5feb36ed603276a9684ea240874a504dada69670c0f83a9c8767e9744e24
a24c92dd02ca1aa94c83430d748db81ed415ac4c9b3e66593b4b2f15b094ca42a1abf6e4e9b17cba21162c
10450c9d7314ff2ae8b62c32133c749d1d9d292d6fd320837b449a7d51a798b74b3e91cf542dc623fa
Signature algorithm: SHA256WithRSASignature
Key size: 2048
Issuer: CN=crypt
Subject: CN=crypt
Valid from: Feb 06 21:51:31 2020 EST (-0500)
Valid until: Apr 24 22:51:31 2028 EDT (-0400)

```

If the RKM server is running with a certificate chain from a CA, follow these steps:

1. Manually copy the files of the certificate chain from the server to a location that is accessible to the key client.

Note: If you have not already done so, save the files of the certificate chain to a secure location. Include the root certificate file, any intermediate certificate files, and the endpoint certificate file. Now, when a client certificate expires, you do not need to download the certificate chain from the

server again. You can add your local copy of the files in the server certificate chain to the new client keystore. For more information, see [“Renewing expired client certificates” on page 845](#).

2. For each certificate in the chain, do the following actions:

- Issue the **mmgskkm print** command to display the contents of the certificate. The following example displays the first certificate of a chain:

```
mmgskkm print --cert sklmChain0.cert
```

- In the **mmgskkm print** command output, find the expiration date for the server certificate. In the following example, the expiration date is on the final line, which begins "Valid until":

```
# mmsklmconfig restcert --host hs21n62 --port 5696 --prefix sklmCert --keystore
sklm4Client.p12 --keypass keystorePass
# ls -ltr sklmCert0.cert
-rw-r--r--. 1 root root 1017 Jul 27 00:55 sklmCert0.cert
# mmgskkm print --cert sklmCert0.cert
Serial number: 2f2409efce9447
SHA-256 digest: 0c9fabf65ab3bea6259af4829cf4027db1395d46a71d49631af7c2a3454ff20d
Signature:
788c8c9a3ec673ac7276283f6720ff4c910f9235042f2959eb37a466277d11a9f085112e28126b05c6451
6
50c9595bd21ab48aabac1ac1fab4a8e945f3dfd2de12c82f57c44e13d983305c3a7ba41d8d565c9db6a54
5
981c16b12af7538f85740e6d0500266cec9fc2cf4b878c7ef12d18fd10e43c0933d246ab825dc5f059c6b
b
0e82f5fabd302e661584deb63b5feb36ed603276a9684ea240874a504dada69670c0f83a9c8767e9744e2
4
a24c92dd02ca1aa94c83430d748db81ed415ac4c9b3e66593b4b2f15b094ca42a1abf6e4e9b17cba21162
c
10450c9d7314ff2ae8b62c32133c749d1d9d292d6fd320837b449a7d51a798b74b3e91cf542dc623fa
Signature algorithm: SHA256WithRSASignature
Key size: 2048
Issuer: CN=crypt
Subject: CN=crypt
Valid from: Feb 06 21:51:31 2020 EST (-0500)
Valid until: Apr 24 22:51:31 2028 EDT (-0400)
```

Key client certificate

If the client certificate file is available, issue the **mmgskkm print** command to display the contents of the client certificate. In the following example, the expiration date is on the final line, which begins "Valid until":

```
# mmgskkm print --cert dsm64Client.cert
Serial number: 3c4e5eae7b9785ec
SHA-256 digest: 2f97b01a1ac82b05cbdc1ac9dfe925cdb03afbddd196d8e312068923c08ceaa36
Signature:
a51f8c10d5970e96eda2b8394b334d51886b827d05585edf222c881410e5cbceff4023281f5b5b9aebb4b
357afb56909b9d070c9fb971c5fdf5436d22526e8903a7f663da8f7380c85b31e23f48e551c9c366edc3bc
331f6b146c6908e50aca0a69432f1cd5f130eec5afaeeb2ef85bdd9d474345719bfc2c82c23bf96066f4ec
80d3ea43986297ace819435b547d7685c81b786d6ffacd2b0a6f6842502b5641f44dbf8acf90cb82e59595
d1f5bb83466f7ce573d290eab76e2cbc9401017f0155a0150a7c12442b68aa4ec403f0f448ff3112039721
85a3f39932aea84847266b9931156660bc3286153d4064e2eda29068661ef298c1cd6a3735f50a02e7
Signature algorithm: SHA256WithRSASignature
Key size: 2048
Issuer: CN=dsm64Client
Subject: CN=dsm64Client
Valid from: Mar 08 18:08:15 2020 EDT (-0400)
Valid until: Mar 09 17:08:15 2021 EST (-0500)
```

If the client certificate file is not available, issue the **openssl** command to extract the client certificate from the client keystore and show the expiration date. In the following example, the client keystore file is SKLM.p12:

```
# openssl pkcs12 -in SKLM.p12 -nodes | openssl x509 -noout -enddate
Enter Import Password:
```

```
MAC verified OK
notAfter=Jun 12 04:10:16 2028 GMT
```

For more information, see [OpenSSL](#).

Error message for an expired RKM server certificate

When the certificate of an RKM server expires, IBM Storage Scale can no longer retrieve master encryption keys (MEKs) from the server. The result is that attempts to create, open, read, or write encrypted files fail with an "Operation not permitted" error. Each time that an error occurs, IBM Storage Scale writes error messages like the following ones to the `/var/adm/ras/mmfs.log.latest` log file:

```
[W] The key server sklm1 (port 5696) had a failure and will be
quarantined for 1 minute(s).
[E] Unable to create encrypted file testfile.enc (inode 21260,
fileset 0, file system gpfs1).
[E] Key 'KEY-uuid:sklm1' could not be fetched. Bad certificate.
```

Error message for an expired key client certificate

IBM Storage Scale checks the status of a key client certificate each time it loads a keystore. It loads a keystore whenever a file system is mounted, or a new policy is applied, or an `RKM.conf` configuration file is explicitly loaded with the `tsloadikm run` command.

When IBM Storage Scale detects an expired client certificate, it writes one or more of the following error messages. The messages are written to the `/var/adm/ras/mmfs.log.latest` log file or to the console or to both, depending on the action that you took just before the problem occurred.

```
[E] Error while validating policy 'policy.enc': rc=778:
While parsing file '/var/mmfs/etc/RKM.conf':
[E] Certificate with label 'GPFSlabel' for backend 'sklm2' has expired.
```

Renewing expired server certificates

Follow these instructions to renew expired server certificates for the simplified setup, the regular setup, and certificate chains.

See the following topics for detailed instructions:

- “[Creating a server certificate](#)” on page 838
- “[Simplified setup: Trusting a new self-signed SKLM server certificate](#)” on page 839
- “[Simplified setup: Trusting a new SKLM server certificate chain](#)” on page 839
- “[Simplified setup: Trusting a new SKLM WebSphere Application Server certificate](#)” on page 840
- “[Regular setup: Trusting a new self-signed SKLM server certificate](#)” on page 840
- “[Trusting a new endpoint server certificate in a server certificate chain](#)” on page 842
- “[Regular setup: Trusting a new SKLM server certificate chain](#)” on page 842
- “[Trusting a new DSM server certificate chain](#)” on page 843

Creating a server certificate

The steps for creating a new server certificate to replace one that is expired are similar to the steps for creating an initial server certificate. Follow the instructions in the documentation of your Remote Key Manager (RKM), which must be one of the following products:

- IBM Security Key Lifecycle Manager (SKLM)
- Thales Vormetric Data Security Manager (DSM)

For more information, see the section *Establishing an encryption-enabled environment* in the *IBM Storage Scale: Administration Guide*.

Simplified setup: Trusting a new self-signed SKLM server certificate

These instructions assume that you are using the simplified setup method and that you have created a new self-signed SKLM server certificate and set it as the in-use certificate. For more information about creating a new SKLM certificate, see step 7 in “[Part 1: Installing and configuring SKLM](#)” on page 749.

1. Find the name of the key server object that needs to be updated. To display a list of the available key server objects, issue the following command on the IBM Storage Scale command line:

```
mmkeyserv server show
```

2. Issue the following command to update the server certificate of the key server object:

```
mmkeyserv server update <serverName>
```

The variable `<serverName>` is the name of the key server object that you want to update.

3. Enter the `SKLMAdmin` administrator password when prompted.
4. Enter yes to trust the SKLM REST certificate.

The key server object is updated with the self-signed server certificate.

Simplified setup: Trusting a new SKLM server certificate chain

These instructions assume that you are using the simplified setup method and you have a certificate chain from a CA. The certificate chain contains a renewed intermediate certificate or a renewed root certificate. For information about obtaining a certificate chain from a CA, see the subtopic [Part 2: Configuring SKLM](#) in “[Simplified setup: Using SKLM with a certificate chain](#)” on page 761.

1. Find the name of the key server object that needs to be updated. To display a list of the available key server objects, issue the following command on the IBM Storage Scale command line:

```
mmkeyserv server show
```

2. Set up the files in the certificate chain by performing the following steps:

- a. Copy the files for the new server certificate chain into the same directory.
- b. Rename each certificate file with the same prefix, followed by a numeral that indicates the order of the certificate in the chain, followed by the file extension .cert. Start the numbering with 0 for the root certificate. For example, if the chain consists of three certificate files and the prefix is `sklmChain`, rename the files as follows:

```
sklmChain0.cert  
sklmChain1.cert  
sklmChain2.cert
```

If the certificate chain contains more than three certificate files, combine the intermediate files into one certificate file, set the numeral in the name of the combined certificate file to 1, and set the numeral in the name of the endpoint certificate file to 2. For example, the certificate chain contains four certificate files: `sklmChain0.cert`, `sklmChain1.cert`, `sklmChain2.cert`, and `sklmChain3.cert`.

3. Issue the following command to update the server certificate of the key server object:

```
mmkeyserv server update <serverName> --kmip-cert sklmChain
```

The variable `<serverName>` is the name of the key server object that you want to update.

4. Enter the `SKLMAdmin` administrator password when prompted.
5. Enter yes to trust the certificate chain.

Simplified setup: Trusting a new SKLM WebSphere Application Server certificate

These instructions assume that you are using the simplified setup method with IBM WebSphere Application Server and SKLM.

1. The simplified setup communicates with SKLM on both the KMIP port and the REST administration port.

On the REST port, the server certificate is the one that is configured in WebSphere Application Server. SKLM runs on WebSphere Application Server.

2. Find the name of the IBM Storage Scale key server object that is associated with SKLM on the REST port. To see a list of key server objects, issue the following command:

```
mmkeyserv server show
```

3. Issue the following command to update the key server object with the new WebSphere Application Server certificate:

```
mmkeyserv server update <serverName>
```

The variable *<serverName>* is the name of the key server object that you want to update.

4. Enter the SKLMAuth administrator password when prompted.

5. Enter yes to trust the SKLM REST certificate.

The IBM Storage Scale client now trusts the new SKLM WebSphere Application Server certificate.

Regular setup: Trusting a new self-signed SKLM server certificate

Follow these instructions if you are using IBM Storage Scale 4.1.1 or later. These instructions assume that you are using SKLM and the regular setup method and that you have created a self-signed SKLM server certificate.

1. Get information about the key client from the /var/mmfs/etc/RKM.conf file:

a) Open the file and find the RKM stanza for the key client that you want to configure.

b) Make a note of the following information from the RKM stanza:

- The password for the client keystore and client certificate, which is specified by the **passphrase** term. You need this information for Step 2.
- The path and file name of the client keystore, which is specified by the **keyStore** term. You need this information for Step 3.

2. Store the client keystore password from Step 1 into a text file, such as /root/keystore.pwd, that is accessible only by the root user.

3. Issue the **mmsklmconfig** command to retrieve the new self-signed SKLM server certificate. This command is available in IBM Storage Scale 4.2.1 and later.

The command connects to the KMIP port, waits for the TLS handshake, and retrieves the certificate that the server presents.

```
mmsklmconfig restcert --host <sklmhost> --port <kmipport>
--prefix <sklmChain> --keystore <rkmKeystore>
--keypass <rkmPassfile> --fips <fips> --nist <nist>
```

The command specifies the following parameters:

--host <sklmhost>

Is the IP address or host name of the RKM server.

--port <kmipport>

Is the KMIP port number of the SKLM server. The default value is 5696.

--prefix <sklmChain>

Is the path and file name prefix where the server certificate files are to be stored.

--keystore <rkmKeystore>

Is the path and file name of the client keystore from Step 1.

--keypass <rkmPassfile>

Is the path and file name of the keystore password file from Step 2.

--fips <fips>

Indicates whether the IBM Storage Scale cluster is using FIPS 140-2-compliant cryptographic modules. Valid values are **on** or **off**. Enter the following command to determine the state:

```
mmlsconfig FIPS1402mode
```

--nist <nist>

Indicates whether the IBM Storage Scale cluster is using encryption that is in compliance with NIST SP800-131A recommendations. Valid values are **on** or **off**. Enter the following command to determine the state:

```
mmlsconfig nistCompliance
```

4. Optional: Display the contents of the retrieved server certificate file and verify that the information matches the information in the new server certificate on the RKM server.

The **mmgskkm** command is available in IBM Storage Scale 4.2.1 and later. Issue the following command:

```
mmgskkm print --cert sklmChain0.cert
```

where *sklmChain* is the path and file name prefix of the certificate files. You specified this prefix in Step 3.

5. Issue the following command to add the retrieved server certificate to the client keystore:

The **mmgskkm** command is available in IBM Storage Scale 4.2.1 and later.

```
mmgskkm trust --prefix <sklmChain> --out <rkmKeystore> --pwd-file <rkmPassfile>
--label <serverLabel>
```

The command specifies the following parameters:

--prefix <sklmChain>

Is the path and file name prefix of the server certificate files. You specified this prefix in Step 3.

--out <rkmKeystore>

Is the path and file name of the client keystore from Step 1.

--pwd-file <rkmPassfile>

Is the path and file name of the client keystore password file that you created in Step 2.

--label <serverLabel>

Is the label under which to store the server certificate in the client keystore.

Note: The label must be unique in the keystore. In particular, it cannot be the label of the expired server certificate from the SKLM key server.

6. Copy the updated client keystore file to all the nodes in the IBM Storage Scale cluster.

7. Reload the new client keystore by one of the following methods:

- On any administration node in the cluster, run the **mmchpolicy** command to refresh the current policy rules. You do not need to repeat this action on other nodes in the cluster.
- On each node of the cluster, unmount and mount the file system.
- In IBM Storage Scale 4.2.1 and later, issue the following command on each node of the cluster:

```
/usr/lpp/mmfs/bin/tsloadikm run
```

The IBM Storage Scale client now trusts the new self-signed SKLM server certificate.

Trusting a new endpoint server certificate in a server certificate chain

These instructions assume that the certificate chain includes a root certificate that is signed by a certificate authority (CA), zero or more intermediate certificates, and an endpoint certificate.

1. If only the endpoint certificate expired and was renewed, you do not need to take any further action on the client side.

This situation occurs, for example, in DSM when you renew an endpoint certificate by running the **gencert** command.

2. If an intermediate certificate or the root certificate expired and was renewed, follow the instructions in one of the following two subtopics:

["Regular setup: Trusting a new SKLM server certificate chain" on page 842](#)

["Trusting a new DSM server certificate chain" on page 843](#)

Regular setup: Trusting a new SKLM server certificate chain

These instructions assume that you are using SKLM and the regular setup method and that you have a certificate chain from a CA. The certificate chain contains a renewed intermediate certificate or a renewed root certificate. For information about obtaining a certificate chain from a CA, see the subtopic "Part 2: Configuring SKLM" in ["Regular setup: Using SKLM with a certificate chain" on page 797](#).

1. Get the path and password of the keystore file of the key client that you are configuring:
 - a) Open the `/var/mmfs/etc/RKM.conf` file and find the RKM stanza of the key client.
 - b) Make a note of the following items:
 - The password for the client keystore and client certificate, which is specified by the **passphrase** term. You need this information for Step 2.
 - The path and file name of the client keystore, which is specified by the **keyStore** term. You need this information for Step 5.
2. Store the client keystore password from Step 1 into a text file, such as `/root/keystore.pwd`, that is accessible only by the root user.
3. Set up the files in the certificate chain:
 - a) Copy the files for the new server certificate chain into the same directory in which the `keystore.pwd` file is located.
 - b) Rename each certificate file with the same prefix, followed by a numeral that indicates the order of the certificate in the chain, followed by the file extension `.cert`. Start the numbering with 0 for the root certificate. For example, if the chain consists of three certificate files and the prefix is `sklmChain`, rename the files as follows:

```
sklmChain0.cert  
sklmChain1.cert  
sklmChain2.cert
```

If the certificate chain contains more than three certificate files, combine the intermediate files into one certificate file, set the numeral in the name of the combined certificate file to 1, and set the numeral in the name of the endpoint certificate file to 2. For example, suppose that the certificate chain contains four certificate files: `sklmChain0.cert`, `sklmChain1.cert`, `sklmChain2.cert`, and `sklmChain3.cert`. Modify the certificate files in the following way:

- The `sklmChain0.cert` file needs no changes.
- Combine `sklmChain1.cert` and `sklmChain2.cert` into one file and name it `sklmChain1.cert`.
- Rename `sklmChain3.cert` to `sklmChain2.cert`.

Important: If you have not already done so, save the files of the certificate chain to a secure location. Include the root certificate file, any intermediate certificate files, and the endpoint certificate file. Now, when a client certificate expires, you will not need to download the certificate chain from the server again. You can add your local copy of the files in the server certificate chain

to the new client keystore. For more information, see “[Renewing expired client certificates](#)” on page 845.

4. Optional: You can verify the server certificate chain by issuing the **openssl verify** command. The command has the following usage:

```
openssl verify -CAfile <rootCaCert> [-untrusted <intermediateCaCerts>] <endpointCert>
```

where:

-CAfile <rootCaCert>

Specifies the root certificate file.

-untrusted <intermediateCaCerts>

Specifies the file that contains the intermediate certificates. If the chain has more than one intermediate certificate, you must combine them into a single file. If the chain has no intermediate certificates, omit this parameter.

<endpointCert>

Specifies the endpoint certificate file.

For example, if your server certificate chain consists of the three sample files that are listed in Step 3, issue the following command:

```
openssl verify -CAfile /root/sk1mChain0.cert -untrusted /root/sk1mChain1.cert /root/sk1mChain2.cert
```

5. Issue the following command to add the new SKLM server certificate chain to the keystore.

The **mmgskkm** command is available in IBM Storage Scale 4.2.1 and later.

```
mmgskkm trust --prefix <sklmChain> --out <keystore> --pwd-file <pwd-file> --label <serverLabel>
```

where:

--prefix <sklmChain>

Is the path and file name prefix of the certificate chain files that you set up in Step 3, such as /root/sk1mChain.

--out <keystore>

Is the path and file name of the client keystore from Step 1.

--pwd-file <pwd-file>

Is the path and file name prefix of the keystore password file that you created in Step 2.

--label <serverLabel>

Is the label under which to store the server certificate in the client keystore.

Note: The label must be unique in the keystore. Also, it cannot be the label of the expired server certificate from the SKLM key server.

6. Copy the updated client keystore to all nodes in the IBM Storage Scale cluster.

7. Reload the new client keystore by one of the following methods:

- On any administration node in the cluster, run the **mmchpolicy** command to refresh the current policy rules. You do not need to repeat this action on other nodes in the cluster.
- On each node of the cluster, unmount and mount the file system.
- In IBM Storage Scale 4.2.1 and later, issue the following command on each node of the cluster:

```
/usr/lpp/mmfs/bin/tsloadikm run
```

The IBM Storage Scale client now trusts the new SKLM server certificate chain.

Trusting a new DSM server certificate chain

These instructions assume that you are using DSM and that you have a DSM certificate chain that you renewed by running the **security genca** command.

- Get the path and password of the keystore file of the key client that you are configuring:
 - Open the /var/mmfs/etc/RKM.conf file and find the RKM stanza of the key client.
 - Make a note of the following items:
 - The password for the client keystore and client certificate, which is specified by the **passphrase** term. You need this information for Step 2.
 - The path and file name of the client keystore, which is specified by the **keyStore** term. You need this information for Step 5.
- Store the client keystore password from Step 1 into a text file, such as /root/keystore.pwd, that is accessible only by the root user.
- Issue the **mmsk1mconfig** command to retrieve the new self-signed DSM server certificate chain. This command is available in IBM Storage Scale 4.2.1 and later.

The command connects to the KMIP port, waits for the TLS handshake, and retrieves the certificate that the server presents.

```
mmsk1mconfig restcert --host <dsmhost> --port <dsmport>
--prefix <dsmChain> --keystore <rkmKeystore>
--keypass <rkmPassfile> --fips <fips> --nist <nist>
```

The command specifies the following parameters:

--host <dsmhost>

Is the IP address or host name of the DSM server.

--port <dsmport>

Is the port number of the DSM web GUI. The default value is 8445.

--prefix <sklmChain>

Is the path and file name prefix where the server certificate files are to be stored.

--keystore <rkmKeystore>

Is the path and file name of the client keystore from Step 1.

--keypass <rkmPassfile>

Is the path and file name of the keystore password file from Step 2.

--fips <fips>

Indicates whether the IBM Storage Scale cluster is using FIPS 140-2-compliant cryptographic modules. Valid values are **on** or **off**. Enter the following command to determine the state:

```
mmlsconfig FIPS1402mode
```

--nist <nist>

Indicates whether the IBM Storage Scale cluster is using encryption that is in compliance with NIST SP800-131A recommendations. Valid values are **on** or **off**. Enter the following command to determine the state:

```
mmlsconfig nistCompliance
```

DSM server certificate chain: The DSM server certificate chain typically consists of two certificates, a DSM internal root CA certificate and an endpoint certificate. The names of certificate files that you retrieve in this step have the following format: the path and file name prefix that you specify in the **--prefix** parameter, followed by a 0 for the root certificate or a 1 for the endpoint certificate, followed by the suffix .cert. In the following example, the prefix is /root/dsmChain:

```
/root/dsmChain0.cert
/root/dsmChain1.cert
```

- Optional: Display the contents of the retrieved server certificate files and verify that the information matches the information in the new server certificate on the DSM server.

The **mmgskkm** command is available in IBM Storage Scale 4.2.1 and later. Issue the following commands:

```
mmgskkm print --cert <dsmChain>0.cert  
mmgskkm print --cert <dsmChain>1.cert
```

where *dsmChain* is the path and file name prefix of the certificate files that you retrieved in Step 3.

5. Issue the following command to add the new DSM server certificate chain to the client keystore.

The **mmgskkm** command is available in IBM Storage Scale 4.2.1 and later.

```
mmgskkm trust --prefix <dsmChain> --out <rkmKeystore> --pwd-file <rkmPassfile>  
--label <serverLabel>
```

The command has the following parameters:

--prefix <dsmChain>

Is the path and file name prefix of the certificate chain files that you retrieved in Step 3, such as /root/dsmChain.

--out <rkmKeystore>

Is the path and file name of the client keystore from Step 1.

--pwd-file <rkmPassfile>

Is the path and file name prefix of the keystore password file that you created in Step 2.

--label <serverLabel>

Is the label under which to store the server certificate in the client keystore.

Note: The label must be unique in the keystore. Also, it cannot be the label of the expired server certificate from the DSM key server.

6. Copy the updated client keystore to all nodes in the IBM Storage Scale cluster.

7. Reload the new client keystore by one of the following methods:

- On any administration node in the cluster, run the **mmchpolicy** command to refresh the current policy rules. You do not need to repeat this action on other nodes in the cluster.
- On each node of the cluster, unmount and mount the file system
- In IBM Storage Scale 4.2.1 and later, issue the following command on each node of the cluster:

```
/usr/lpp/mmfs/bin/tsloadikm run
```

The IBM Storage Scale client now trusts the new self-signed DSM server certificate.

Renewing expired client certificates

Follow these instructions to create and renew expired client certificates for the simplified setup, the regular setup, or the setup for Thales Vormetric Data Security Manager (DSM).

See the following subtopics:

[“Simplified setup: Updating a key client certificate \(5.1.0 or later\)” on page 846](#)

[“Simplified setup: Updating a key client certificate \(5.0.5\)” on page 848](#)

[“Simplified setup: Updating a key client certificate \(earlier than 5.0.5\)” on page 848](#)

[“Regular setup or DSM setup: Creating and installing a new key client” on page 850](#)

[“Regular setup: Trusting a new client certificate” on page 855](#)

[“DSM: Trusting a new client certificate” on page 855](#)

Note: For more information about the simplified setup method, see the following topics:

- [“Simplified setup: Using SKLM with a self-signed certificate” on page 747](#)

- [“Simplified setup: Using SKLM with a certificate chain” on page 761](#)

For more information about the regular setup method or the setup for DSM, see the following topics:

- [“Regular setup: Using SKLM with a self-signed certificate” on page 788](#)

- “Regular setup: Using SKLM with a certificate chain” on page 797
- “Configuring encryption with the Thales Vormetric DSM key server” on page 812

Note: IBM Storage Scale supports IBM Security Guardium Key Lifecycle Manager (GKLM) 4.1.0.1 (IF01), 4.1.1, or later. The older versions of GKLM are referred to as IBM Security Lifecycle Manager or SKLM in the documentation. The configuration information is the same for both GKLM and SKLM.

Simplified setup: Updating a key client certificate (5.1.0 or later)

Follow these instructions if you are using the simplified setup method and the key client is running IBM Storage Scale 5.1.0 or later.

To update an expired or unexpired key client certificate, follow these steps.

- Issue the **mmkeyserv client show** command and verify the type of the current client certificate. If the type is system-generated, then the certificate is a self-signed certificate that was generated by the system. If the type is user-provided, the certificate is a CA-signed certificate. In the following example, the final line indicates that the certificate is user-provided:

```
# mmkeyserv client show
c34f2n03Client1
  Label: c6f2bc3n9client
  Key Server: keyserver01.gpfs.net
  Tenants: devG1
  Certificate Expiration: 2020-09-04 22:40:41 (-0400)
  Certificate Type: user-provided
```

- If the certificate type is system-generated, go to Step 2.
 - If the certificate type is user-provided, go to Step 3.
- This step describes the actions to take if the type of the certificate to be replaced is system-generated:

- If you want to replace the current system-generated self-signed certificate with another system-generated self-signed certificate, issue the **mmkeyserv client update** command with an expire time for the new certificate. You must specify a password file that contains the key server password, an expire time, and a password file that contains a new password for the client keystore. In the following example the expire time is 90 days:

```
# mmkeyserv client update c6f2bc3n9client --server-pwd /root/c6f2bc3n9.pw --days 90 --
keystore-pwd /root/client.pw
mmkeyserv: [I] Client currently does not have access to the key. Continue the
registration process ...
mmkeyserv: Successfully accepted client certificate
mmkeyserv: Propagating the cluster configuration data to all affected nodes. This is
an asynchronous process.
mmkeyserv: Deleting the following KMIP certificate with label:
2454534160085868372_vut_1578295912
Fri Sep 18 12:01:33 EST 2020: mmcommon pushSdr_async: mmsdrfs propagation started
# Fri Sep 18 12:01:36 EST 2020: mmcommon pushSdr_async:
mmsdrfs propagation completed; mmdsh rc=0
```

- If you want to replace the current system-generated self-signed certificate with a CA-signed certificate, issue the **mmkeyserv client update** command and specify the CA-signed certificate information. In the following example, the client CA-signed certificate file is /tmp/client.cert, the client private key file is /tmp/client.key, and the CA certificate chain file is /tmp/CA-chain:

```
# mmkeyserv client update c6f2bc3n9client --cert /tmp/client.cert --priv /tmp/client.key
--ca-chain /tmp/CA-chain --server-pwd /root/c6f2bc3n9.pw --keystore-pwd /root/client.pw
mmkeyserv: [I] Client currently does not have access to the key. Continue the registration
process ...
mmkeyserv: Successfully accepted client certificate
mmkeyserv: Propagating the cluster configuration data to all affected nodes. This is an
asynchronous process.
mmkeyserv: Deleting the following KMIP certificate with label:
2454534160085868372_vut_1600467545
Fri Sep 18 12:19:35 EST 2020: mmcommon pushSdr_async: mmsdrfs propagation started
(12:19:38) c9f1u19p1:~ # Fri Sep 18 12:19:38 EST 2020: mmcommon pushSdr_async:
mmsdrfs propagation completed; mmdsh rc=0
```

The **mmkeyserv client update** command deregisters the key client with the current certificate from any tenants that it is registered to and deletes the key client. The command then creates a new key client with the same name as the old one, creates a new self-signed client certificate or uses the new CA-signed client certificate, and stores the new client certificate into the client keystore. Finally, the command registers the new key client with any tenants that the old key client was registered to.

Go to Step 4.

3. This step describes the actions to take if the type of the certificate to be replaced is a user-provided CA-signed certificate:

- If you want to replace the current user-provided CA-signed certificate with a system-generated self-signed certificate, you must issue the **mmkeyserv client update** command with the **--force** option. The following example is exactly like the command that is used to replace the system-generated self-signed certificate in Step 2, except that here you must specify the **--force** option because you are replacing a CA-signed certificate:

```
# mmkeyserv client update c6f2bc3n9client --server-pwd /root/c6f2bc3n9.pw --days 90 --keystore-pwd /root/client.pw --force
    mmkeyserv: [I] Client currently does not have access to the key. Continue the registration process ...
        mmkeyserv: Successfully accepted client certificate
        mmkeyserv: Propagating the cluster configuration data to all affected nodes. This is an asynchronous process.
        mmkeyserv: Deleting the following KMIP certificate with label:
2454534160085868372_vut_1578295912
    Fri Sep 18 12:01:33 EST 2020: mmcommon pushSdr_async: mmsdrfs propagation started
# Fri Sep 18 12:01:36 EST 2020: mmcommon pushSdr_async:
    mmsdrfs propagation completed; mmdsh rc=0
```

If you do not specify the **--force** option, the command displays an error message, as in the following example:

```
#mmkeyserv client update c6f2bc3n9client --server-pwd /root/c6f2bc3n9.pw --days 90 --
keystore-pwd /root/client.pw
    mmkeyserv: Client c6f2bc3n9client was initially created using a user-provided certificate.
        Provide a new client certificate with the --cert option or use the --force option to generate a new self-signed certificate for the client.

    mmkeyserv: Command failed. Examine previous error messages to determine cause.
```

- If you want to replace the current user-provided CA-signed certificate with another CA-signed certificate, issue the **mmkeyserv client update** command and specify the CA-signed certificate information. You do not have to specify the **--force** option because you are replacing the CA-signed certificate with another CA-signed certificate. In the following example, the client CA-signed certificate file is `/tmp/client.cert`, the client private key file is `/tmp/client.key`, and the CA certificate chain file is `/tmp/CA-chain`:

```
# mmkeyserv client update c6f2bc3n9client --cert /tmp/client.cert --priv /tmp/client.key --ca-chain /tmp/CA-chain --server-pwd /root/c6f2bc3n9.pw --keystore-pwd /root/client.pw
    mmkeyserv: [I] Client currently does not have access to the key. Continue the registration process ...
        mmkeyserv: Successfully accepted client certificate
        mmkeyserv: Propagating the cluster configuration data to all affected nodes. This is an asynchronous process.
        mmkeyserv: Deleting the following KMIP certificate with label:
2454534160085868372_vut_1600467545
    Fri Sep 18 12:19:35 EST 2020: mmcommon pushSdr_async: mmsdrfs propagation started
(12:19:38) c9f1u19p1:~ # Fri Sep 18 12:19:38 EST 2020: mmcommon pushSdr_async:
    mmsdrfs propagation completed; mmdsh rc=0
```

The **mmkeyserv client update** command does the same actions that are described in the last paragraph of Step 2.

4. Issue the **mmkeyserv client show** command and verify the new client certificate expiration date.

For more information, see the topic *mmkeyserv command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

The new client certificate is accepted by the SKLM key server. The new key client is registered with any tenants that the old key client was registered to.

Simplified setup: Updating a key client certificate (5.0.5)

Follow these instructions if you are using the simplified setup method and the key client is running IBM Storage Scale 5.0.5.

To update an expired or unexpired key client certificate, follow these steps:

1. Issue the **mmkeyserv client show** command and verify the information that is displayed about the key client whose certificate is expired. In the following example only one key client exists in the cluster:

```
# mmkeyserv client show
c34f2n03Client1
  Label:          c6f2bc3n9client
  Key Server:    keyserver01.gpfs.net
  Tenants:       devG1
  Certificate Expiration: 2020-10-01 14:20:46 (-0400)
```

2. Issue the **mmkeyserv client update** command to create a new client certificate to replace the existing one. The command deregisters the key client with the existing certificate from any tenants that it is registered to and deletes the key client. The command then creates a new key client with the same name as the old one, creates a new client certificate, and stores the new client certificate into the client keystore. Finally, the command registers the new key client with any tenants that the old key client was registered to.

In the following example, the command provides the following a password file that contains the key server password. It also specifies an expire time of 90 days for the new client certificate and provides a password file that contains a new password for the client keystore:

In the following example,

```
# mmkeyserv client update c6f2bc3n9client --server-pwd /u/admin/README/sklm/c6f2bc3n9.pw
  --days 90 --keystore-pwd /u/admin/README/sklm/client.pw
  mmkeyserv: [I] Client currently does not have access to the key. Continue the
registration
  process ...
  mmkeyserv: Successfully accepted client certificate
  mmkeyserv: Propagating the cluster configuration data to all affected nodes. This is an
asynchronous process.
  mmkeyserv: Deleting the following KMIP certificate with label:
  2454534160085868372_vut_1578295912
  Mon Jan 6 12:06:33 EST 2020: mmcommon pushSdr_async: mmsdrfs propagation started
  (12:06:35) c9f1u19p1:~ # Mon Jan 6 12:06:36 EST 2020: mmcommon pushSdr_async:
  mmsdrfs propagation completed; mmdsh rc=0
```

3. Issue the **mmkeyserv client show** command and verify the new client certificate expiration date.

For more information, see the topic *mmkeyserv command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

The new client certificate is accepted by the SKLM key server. The new key client is registered with any tenants that the old key client was registered to.

Simplified setup: Updating a key client certificate (earlier than 5.0.5)

Follow these instructions if you are using the simplified setup method and the key client is running a version of IBM Storage Scale that is earlier than 5.0.5.

To update an expired or unexpired key client certificate, you must create and register a new key client and deregister the old key client. These instructions assume that you want to create a key client `c1Client1`, deregister the old client `c1Client0`, and register the new key client with tenant `devG1` on key server `keyserver01`.

1. Issue the following command to create the key client. Enter a password and a pass phrase when prompted:

```
# mmkeyserv client create c1Client1 --server keyserver01
Enter password for the key server keyserver01:
Create a pass phrase for keystore:
Confirm your pass phrase:
```

2. Issue the following command to display information about the current clients.

The command output shows that the existing client c1Client0, which has the expired certificate, is registered with tenant devG1 on key server keyserver01. The new client c1Client1 is not registered with a tenant:

```
# mmkeyserv client show
c1Client0
  Label:          c1Client0
  Key Server:    keyserver01
  Tenants:       devG1
  Certificate expiration: 2023-04-22 15:41:21 (-0400)

c1Client1
  Label:          c1Client1
  Key Server:    keyserver01
  Tenants:       (none)
  Certificate expiration: 2023-04-22 15:41:21 (-0400)
```

3. Optional: Issue the following command and make a note of the RKM ID that is associated with the old key client.

It is a good idea to reuse the RKM ID of the old key client when you register the new key client. If you do so, then you do not have to update any of your encryption policy rules that specify the RKM ID:

```
# mmkeyserv tenant show
devG1
  Key Server:      keyserver01.gpfs.net
  Registered Client: c1Client0
  RKM ID:         keyserver01_devG1
```

See Step 5.

4. Issue the following command to deregister the current key client from the tenant. Notice that this command also deletes the expired certificate:

```
# mmkeyserv client deregister c1Client0 --tenant devG1
Enter password for the key server:
Enter password for the key server of client c1Client0:
mmkeyserv: Deleting the following KMIP certificate with label:
15826749741870337947_devG1_1498047851
```

Note: If you deregister a key client whose certificate is not yet expired, you cannot fetch keys until you register a new key client:

5. Issue the following command to register the new key client with tenant devG1 in key server keyserver01.

In the **--rkm-id** parameter, specify a valid RKM ID for the new connection.

Note: Here you can specify the RKM ID of the old key client to avoid having to update encryption policy rules that reference that RKM ID. See Step 3.

```
# mmkeyserv client register c1Client1 --tenant devG1 --rkm-id keyserver01_devG1
Enter password for the key server:
mmkeyserv: [I] Client currently does not have access to the key.
Continue the registration process ...
mmkeyserv: Successfully accepted client certificate
```

For more information, see the topic *mmkeyserv command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

The new client certificate is accepted by the SKLM key server. The new key client is registered with tenant devG1.

Regular setup or DSM setup: Creating and installing a new key client

Follow these instructions if you are using either the regular setup method with an SKLM key server or the regular setup with a DSM key server. To update an expired or unexpired client certificate, follow these steps:

Note: The **mmgskkm** command and the **msklmconfig** command are available in IBM Storage Scale 4.2.1 and later.

1. Create a keystore password file that is accessible only by the root user, such as `/root/keystore.pwd`, and store a password in it.
2. The next step depends on the type of client certificate that you want to use:
 - If you want to create a key client with a system-generated self-signed certificate, go to Step 3.
 - If you want to create a key client with a user-provided CA-signed certificate, go to Step 4.
3. This step describes the actions to take if you are creating key client with a system-generated self-signed certificate.
 - a) Issue the **mmgskkm gen** command to create the new client credentials:

```
mmgskkm gen --prefix <prefix> --cname <cname> --fips <fips>
--nist <nist> --days <validdays> --keylen <keylen>
```

where:

--prefix <prefix>

Is the path and file name prefix of the new certificate files and keystore file.

--cname <cname>

Is the name of the new IBM Storage Scale key client. The name can be up to 54 characters in length and can contain alphanumeric characters, hyphen (-), and period (.). In DSM, names are not case-sensitive, so it is a good practice not to include uppercase letters.

--fips <fips>

Is the current value of the **FIPS1402mode** configuration variable in IBM Storage Scale. Valid values are *yes* and *no*. Issue the following command to see the current value:

```
mmlsconfig FIPS1402mode
```

--nist <nist>

Is the current value of the **nistCompliance** configuration variable in IBM Storage Scale. Valid values are **SP800-131A** and **off**. To see the current value, issue the following command:

```
mmlsconfig nistCompliance
```

--days <validdays>

Is the number of days that you want the client certificate to be valid.

--keylen <keylen>

Is the length in bits that you want for the RSA key that is generated.

- b) Issue the **mmgskkm store** command to create a PKCS#12 keystore and to store the system-generated client certificate and private key into it.

```
mmgskkm store --cert <certFile> --priv <privFile> --label <label> --pwd-file <pwd-file>
--out <keystore>
```

where:

--cert <certFile>

Is the client certificate file that you created in Step 3(a). The name of the file has the format `<prefix>.cert`, where `<prefix>` is the path and file name prefix that you specified in Step 3(a).

--priv <privFile>

Is the private key file that you generated in Step 3(a). The name of the file has the format <prefix>.priv, where <prefix> is the path and file name prefix that you specified in Step 3(a).

--label <label>

Is the label under which the new client certificate is stored in the keystore.

--pwd-file <pwd-file>

Is the path and file name of the keystore password file that you created in Step 1.

--out <keystore>

Is the path and file name of the new PKCS#12 keystore.

Note: It is a good practice to generate the keystore files into the directory /var/mmfs/etc/RKMcerts.

c) Go to Step 5.

4. This step describes the actions to take if you are creating a key client with a user-provided CA-signed certificate.

a) Obtain the CA-signed certificate files from a CA.

b) Issue the **mmgskkm store** command to create a PKCS#12 keystore and to store the user-provided CA-signed client certificate and private key into it.

```
mmgskkm store -cert <certFile> -priv <privFile> { -chain <CACertChainFile> | -prefix <CACertPrefix> } -label <label> -pwd-file <pwdFile> -out <keystore>
```

where:

--cert <certFile>

Is the CA signed client certificate file in base64 encoded PEM format.

--priv <privFile>

Is the client's private key file in base64 encoded PEM format, unencrypted.

{ -chain <CACertChainFile> | -prefix <CACertPrefix> }

Is the client credentials. Choose one of the following parameters:

-chain <CACertChainFile>

Is the CA certificate chain file that contains all the CA certificates, beginning with the root CA certificate and ending with the last intermediate CA certificate that signed the client certificate.

-prefix <CACertPrefix>

Is the full path prefix of the CA certificate files, named <CACertPrefix><index>.cert, where <index> is 0 for the root CA certificate and the last index is the last intermediate CA certificate that signed the client certificate.

--label <label>

Is the label under which the new client certificate is stored in the keystore.

--pwd-file <pwd-file>

Is the path and file name of the keystore password file that you created in Step 1.

--out <keystore>

Is the path and file name of the new PKCS#12 keystore.

For more information, see one of the following topics:

[“Regular setup: Using SKLM with a certificate chain” on page 797](#)

[“Configuring encryption with the Thales Vormetric DSM key server” on page 812](#)

5. The next step depend on the type of certificate that the SKLM server is using:

- If the SKLM server certificate is running with a self-signed certificate, go to Step 6.

- If the SKLM server is running with a CA-signed certificate chain, go to Step 7.
6. This step describes the actions to take if the SKLM server is running with a self-signed certificate.
- Issue the **mmsklmconfig** command to retrieve the server certificate and add it to the client keystore:

```
mmsklmconfig restcert --host <rkmHost> --port <rkmPort> --prefix <serverPrefix> --
keystore <keystore>
--keypass <pwd-file> --fips <fips> --nist <nist>
```

The command includes the following parameters:

--host <rkmHost>

Is the IP address or host name of the RKM server.

--port <rkmPort>

Is the port of the RKM server:

- For SKLM, the port is the KMIP port, which has a default value of 5696.
- For DSM, the port is the web GUI port, which has a default value of 8445.

--prefix <serverPrefix>

Is the path and file name prefix for the server certificate.

--keystore <keystore>

Is the path and file name of the PKCS#12 keystore that you created in Step 3(a).

--keypass <pwd-file>

Is the path and file name of the keystore password file that you created in Step 1.

--fips <fips>

Is the current value of the **FIPS1402mode** configuration variable in IBM Storage Scale. Valid values are *yes* and *no*. Issue the following command to see the current value:

```
mmlsconfig FIPS1402mode
```

--nist <nist>

Is the current value of the **nistCompliance** configuration variable in IBM Storage Scale. Valid values are **SP800-131A** and **off**. To see the current value, issue the following command:

```
mmlsconfig nistCompliance
```

- b) Optional: Issue the following command to display the certificate file that you downloaded in Step 6(a). Verify that the information matches the information that is displayed for the current server certificate in the RKM GUI:

```
mmgskkm print --cert <serverPrefix>0.cert
```

where *serverPrefix* is the path and file name prefix of the server certificate that you specified in Step 6(a).

- c) Issue the following command to add the retrieved certificate chain to the client keystore:

```
mmgskkm trust --prefix <serverPrefix> --out <keystore> --pwd-file <pwd-file>
--label <serverLabel>
```

where:

--prefix <serverPrefix>

Is the path and file name prefix for the RKM certificate chain that you specified in Step 4 (b).

--out<keystore>

Is the path and file name of the client keystore that you created in Step 4(b).

--pwd-file<pwd-file>

Is the path and file name of the keystore password file that you created in Step 1.

--label<serverLabel>

Is the label under which you want to store the RKM certificate chain in the client keystore.

d) Update the RKM stanza for the new client credentials in the /var/mmfs/etc/RKM.conf file.

Make sure that the following values are correct:

- The keyStore term specifies the path and file name of the client keystore that you created in Step 3(b).
- The passphrase term specifies the keystore password from Step 1.
- The clientCertLabel term specifies the label of the new client certificate from Step 3(b).

e) Go to Step 8.

7. This step describes the actions to take if the SKLM server is running with a CA-signed certificate chain.

a) Gather the files of the server certificate chain into location that is accessible to the key client:

- i) If you previously saved the certificate files of the server certificate chain into a secure location, ensure that the server certificate chain files are accessible by the key client node. For more information, see [“Regular setup: Using SKLM with a certificate chain” on page 797](#).

If you did not previously save the files of the server certificate chain, follow these steps:

- Manually copy the files of the server certificate chain from the SKLM server to a location that is accessible from the key client.
- Make backup copies of the server certificate files, in case they are lost or damaged.

- ii) Rename each certificate file with the same prefix, followed by a numeral that indicates the order of the certificate in the chain, followed by the file extension .cert. Start the numbering with 0 for the root certificate. For example, if the chain consists of three certificate files and the prefix is sklmChain, rename the files as follows:

```
sklmChain0.cert  
sklmChain1.cert  
sklmChain2.cert
```

b) Optional: Issue the **openssl** command to display the certificate chain that you downloaded in Step 7(a). Verify that the information matches the information that is displayed for the current server certificate in the RKM GUI. In the following example the chain has three certificate files:

```
openssl verify -CAfile /var/mmfs/etc/RKMcerts/sklmChain0.cert  
-untrusted /var/mmfs/etc/RKMcerts/sklmChain1.cert  
/var/mmfs/etc/RKMcerts/sklmChain2.cert
```

The command has the following parameters:

-CAfile /var/mmfs/etc/RKMcerts/sklmChain0.cert

Specifies the path of the root certificate file.

-untrusted /var/mmfs/etc/RKMcerts/sklmChain1.cert

Specifies the path of the intermediate certificate file.

/var/mmfs/etc/RKMcerts/sklmChain2.cert

Specifies the path of the endpoint certificate file.

Note: If the certificate chain contains more than three certificate files, combine the intermediate files into one certificate file, set the numeral in the name of the combined certificate file to 1, and set the numeral in the name of the endpoint certificate file to 2. For example, suppose that the certificate chain contains four certificate files: sklmChain0.cert, sklmChain1.cert, sklmChain2.cert, and sklmChain3.cert. Before you issue the **openssl** command, do the following steps:

- Make backup copies of these certificate files, in case they are lost or damaged.
- Modify the certificate files in the following way:

- The `sklmChain0.cert` file needs no changes.
- Combine `sklmChain1.cert` and `sklmChain2.cert` into one file and name it `sklmChain1.cert`.
- Rename `sklmChain3.cert` to `sklmChain2.cert`.

Issue the `openssl` command in the same way as in the previous example:

```
openssl verify -CAfile /var/mmfs/etc/RKMcerts/sklmChain0.cert
-untrusted /var/mmfs/etc/RKMcerts/sklmChain1.cert
/var/mmfs/etc/RKMcerts/sklmChain2.cert
```

If the chain contains only two certificate files, omit the `-untrusted` option and issue the `openssl` command in the following way:

```
openssl verify -CAfile /var/mmfs/etc/RKMcerts/sklmChain0.cert
/var/mmfs/etc/RKMcerts/sklmChain1.cert
```

Important: Combining the intermediate files into one certificate file is required only for the `openssl` command. It is not required for the `mmgskkm` command.

- c) Issue the following command to add the retrieved certificate chain to the client keystore:

```
mmgskkm trust --prefix <serverPrefix> --out <keystore> --pwd-file <pwd-file>
--label <serverLabel>
```

where:

--prefix <serverPrefix>

Is the path and file name prefix for the RKM certificate chain that you specified in Step 4 (b).

--out<keystore>

Is the path and file name of the client keystore that you created in Step 4(b).

--pwd-file<pwd-file>

Is the path and file name of the keystore password file that you created in Step 1.

--label<serverLabel>

Is the label under which you want to store the RKM certificate chain in the client keystore.

- d) Update the RKM stanza for the new client credentials in the `/var/mmfs/etc/RKM.conf` file. Make sure that the following values are correct:

- The `keyStore` term specifies the path and file name of the client keystore that you created in Step 4(b).
- The `passphrase` term specifies the keystore password from Step 1.
- The `clientCertLabel` term specifies the label of the new client certificate from Step 4(b).

8. Copy the updated `/var/mmfs/etc/RKM.conf` file and the new client keystore file to all the nodes of the cluster.

9. Reload the new client keystore by one of the following methods:

- On any administration node in the cluster, run the `mmchpolicy` command to refresh the current policy rules. You do not need to repeat this action on other nodes in the cluster.
- On each node of the cluster, unmount and mount the file system.
- In IBM Storage Scale 4.2.1 and later, issue the following command:

```
/usr/lpp/mmfs/bin/tsloadikm run
```

Repeat this action on all the nodes of the cluster.

10. Issue the following command to purge all master encryption keys from the cache of the GPFS daemon:

```
tsctl encKeyCachePurge all
```

This action ensures that subsequent reads and writes to files use the new client credentials.

The new client certificate is installed.

Regular setup: Trusting a new client certificate

Follow these instructions if you are using the regular setup method with an SKLM key server. If you have not created and installed new client credentials, follow the instructions in the preceding subsection [“Regular setup or DSM setup: Creating and installing a new key client” on page 850](#).

Follow these instructions if you are using SKLM and the Regular setup method and you have created and installed new client credentials.

1. Add the new client certificate to the SKLM list of pending certificates:

- a) On the node that you are configuring for encryption, try to create an encrypted file by doing some action that triggers an encryption policy rule.
- b) The attempt fails because SKLM does not yet trust the new client certificate. However, the attempt causes SKLM to add the new client certificate to the list of pending certificates in the SKLM key server.

2. Verify the RKM ID in the error message from Step 2:

- a) Find the RKM ID that is specified in the error message.

In the following example, the RKM ID is keyserver01:

```
# touch /gpfs0/test
touch: cannot touch '/gpfs0/test': Permission denied
# tail -n 2 /var/adm/ras/mmfs.log.latest

2020-07-16_01:28:08.792-0400: [E] Unable to create encrypted file file(inode 18560,
fileset 0, file system fs).
2020-07-16_01:28:08.792-0400: [E] Key 'KEY-ad4f3a9-019465da-edc8-49d4-b183-80ae89635cbc:
sklm3RKM' could not be fetched. Invalid request.
```

- b) Find the RKM ID of the RKM stanza that specifies the new client keystore.

The RKM stanza is in the /var/mmfs/etc/RKM.conf file.

- c) Verify that the RKM ID from the error message matches the RKM ID of the stanza.

3. Verify the pending client certificate in SKLM:

- a) On the main page of the SKLM graphical user interface, click **Pending client device communication certificates**.
- b) In the list of certificates, select the new client certificate and click **View**.
- c) Verify that the certificate matches the new client certificate.
- d) If the certificates match, click **Accept and Trust**.

4. Enter a name for the new certificate and click **Accept and Trust** again.

5. Verify that the server accepts the new client certificate:

- a) On the node that you are configuring for encryption, try to create an encrypted file as you did in step 2(a).
- b) This time the command succeeds and the encrypted file is created.

SKLM trusts the new client certificate.

DSM: Trusting a new client certificate

Follow these instructions if you are using the regular setup method with DSM. If you have not created a new client certificate, follow the instructions in the preceding subsection [“Regular setup or DSM setup: Creating and installing a new key client” on page 850](#).

Follow these instructions if you are using the regular setup with DSM as the key server and you have created a new client certificate and imported its information into the current IBM Storage Scale policy rules.

1. In the DSM web GUI, import the new client certificate into the DSM server.

Provide the path and file name of the certificate file that you created in Step 2 and referenced in Step 3 of the subtopic “[Regular setup or DSM setup: Creating and installing a new key client](#)” on page 850. The path and file name have the format <prefix>.cert, where <prefix> is the path and file name prefix that you specified in Step 2.

2. On the node that you are configuring for encryption, try to create an encrypted file by doing some action that triggers an encryption policy rule. These instructions assume that the file is successfully created.

DSM trusts the new client certificate.

Encryption hints

Find useful hints for working with file encryption.

Testing whether a file is encrypted by IBM Storage Scale

To test whether a file is encrypted by IBM Storage Scale, do one of the following actions:

- In a policy, use the following condition:

```
XATTR('gpfs.Encryption') IS NOT NULL
```

For more information, see “[Extended attribute functions](#)” on page 546.

- On the command line, issue the following command:

```
mmlsattr -L FileName
```

The command displays output as shown in the following example. The line of the output that begins with the label Encrypted indicates whether the file is encrypted:

```
#mmlsattr -L textReport
name:          textReport
metadata replication: 1 max 2
data replication: 1 max 2
immutable:      no
appendOnly:     no
flags:
storage pool name: system
fileset name:    root
snapshot name:
creation time:  Tue Jun 12 15:40:30 2018
Misc attributes: ARCHIVE
Encrypted:      yes
```

For more information, see the topic *mmlsattr command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Secure deletion

Secure deletion of encrypted files includes not only deleting the files from the file system but also deleting the appropriate MEKs from the remote key server (RKM) and from the key cache on each IBM Storage Scale node.

Securely deleting files in a fileset

After files are removed from a fileset with standard operating system commands (such as `unlink` and `rm`), the tenant administrator might decide to securely delete them. For example, suppose that until that

point, the FEKs of all files in the fileset were encrypted with the MEK with key name KEY-old:isklmsrv. To cause the secure deletion of all removed files, the administrator must perform the following steps:

1. Create a new MEK and note its key name (in this example, KEY-new:isklmsrv).
 2. Modify the appropriate encryption policy KEYS statement in the encryption policy to encrypt new files with the new MEK (for example, KEY-new:isklmsrv) instead of the old one (KEY-old:isklmsrv).
 3. Create and apply a migration (rewrapping) policy (CHANGE ENCRYPTION KEYS) to scan all files, unwrap the wrapped FEK entries of files that are wrapped with the old key (KEY-old:isklmsrv), and rewrap them with the new key (KEY-new:isklmsrv); this step ensures that the FEKs of existing files are accessible in the future.
- Tip:** The **mmapplypolicy** command always begins by scanning all of the files in the affected file system or fileset to discover files that meet the criteria of the policy rule. In this example, the criterion is whether the file is encrypted with a FEK that is wrapped with the MEK KEY-old:isklmsrv. If your file system or fileset is very large, you might want to delay running **mmapplypolicy** until a time when the system is not running a heavy load of applications. For more information, see the topic “[Phase one: Selecting candidate files](#)” on page 553.
4. Delete any snapshots that might contain files that are encrypted with the old MEK (KEY-old:isklmsrv).



Warning: You will not be able to delete such snapshots after the old MEK is deleted from the key server.

5. Wait for the **mmapplypolicy** command from Step 3 to complete. Do not begin the next step until the **mmapplypolicy** command from Step 3 is complete.
6. Remove the old key, KEY-old:isklmsrv. This step commits the secure deletion of all files that were previously unlinked (and whose FEKs had therefore not been rewrapped with the new MEK, KEY-new:isklmsrv).
7. On each node that has ever done I/O to a file encrypted with the old key (KEY-old:isklmsrv), issue the following command:

```
/usr/lpp/mmfs/bin/tsctl encKeyCachePurge 'KEY-old:isklmsrv'
```

From this point on, the new key is used for encryption, which is performed transparently to the application.

Note: The **mmdelfs** command does *not* perform any secure deletion of the files in the file system to be deleted. The **mmdelfs** command removes only the structures for the specified file system. To securely delete files, follow these steps:

1. Identify all MEKs currently used to wrap the FEKs of files in the file system to be deleted. If this information is not available through other means, follow these steps to obtain it:
 - a. Issue the **mmlsattr -n gpfs.Encryption** command on all files of the file system.
 - b. Parse the resulting output to extract all the distinct key names of the MEKs that are used.

Note: The following list describes the possible ways that an MEK might be in use in a file system:

- a. The MEK is, or was at some point, specified in an encryption rule in the policy set on the file system.
 - b. An FEK rewrap has been run, rewrapping an FEK with another MEK.
2. Determine whether the identified MEKs were used to wrap FEKs in other file systems.
- WARNING:** If the same MEKs were used to wrap FEKs in other file systems, deleting those MEKs results in irreparable data loss in the other file systems where those MEKs are used. Before you delete such MEKs from the key servers, you must create one or more new MEKs and rewrap the files in the other file systems.
3. After appropriately handling any MEKs that were used to wrap FEKs in other file systems (as explained in the warning), delete the identified MEKs from their RKMs.

Secure deletion and encryption key cache purging

The key servers that store the MEKs know how to manage and securely delete keys. After an MEK is deleted, all files whose FEKs were encrypted with that MEK are no longer accessible. Even if the data blocks corresponding to the deleted files are retrieved, the contents of the file can no longer be reconstructed, since the data cannot be decrypted.

However, if the MEKs are cached for performance reasons (so that they do not need to be fetched from the server each time a file is created or accessed), the MEKs must also be purged from the cache to complete the secure deletion.

You can use the following command to purge a key from the key cache, or to clean the entire cache, of an individual node:

```
/usr/lpp/mmfs/bin/tsctl encKeyCachePurge {Key | all}
```

where:

Key

The key ID of the key that you want purged from the key cache, specified in the *KeyId:RkmId* syntax.

all

Indicates that the entire key cache is to be cleaned.

The scope of this command is limited to the local node and must be run on all nodes that accessed the MEKs you are purging to ensure secure deletion.

Note: The steps for secure deletion and encryption key cache purging are similar to the steps for key rotation. For more information, see “[Key rotation: Replacing master encryption keys](#)” on page 858.



Warning:

- If the steps for secure deletion are not followed carefully, they can result in unrecoverable data loss. Be aware of the following issues:
 - Check other file systems that might contain files that are encrypted with the old MEK. If there are such files, rewrap their FEKs with the new MEK before you delete the old MEK from the RKM server.
 - Test the policy rule by running the **mmapplypolicy** command with the **-I test** option. Check the output to verify that the policy rule is selecting the correct set of files. Also verify that the KEYS statement specifies the correct old MEK and new MEK.
 - To preserve the data in files that are deleted or unlinked from filesets, restore the files (from a backup or snapshot, if available) before you issue the **mmapplypolicy** command. Remember that the **mmapplypolicy** command does not process unlinked files that were deleted from filesets with operating system commands such as **rm** and **unlink**.
- Remember that after the old MEK is deleted from the RKM server, any encrypted data in files whose FEKs are wrapped with the old MEK is unrecoverable:
 - The encrypted data of files in filesets that were accidentally unlinked and therefore did not undergo the rewrapping procedure is not recoverable through relinking. After the old MEK is deleted from the server, it is impossible to access any file whose FEK was not rewrapped.
 - Files in other file systems that are encrypted with a FEK that is wrapped with the old MEK are not recoverable.

Key rotation: Replacing master encryption keys

Key rotation is the process of rewrapping file encryption keys (FEKs) with a new master encryption key (MEK).

Replacing a MEK can require rewrapping the FEKs of a large number of files across multiple file systems and also possibly in archives. Before you begin the process of replacing a MEK, back up the affected files in case you need to redo the process.



Warning: If you plan to delete the MEK from the RKM server, be aware that after the MEK is deleted, any files that are still encrypted with FEKs that are wrapped with the old MEK cannot be decrypted and their data is unrecoverable.

1. Create a MEK on the key server and make a note of its key UUID. In this help topic, the keyname of the old MEK is KEY-old:isklmsrv and the keyname of the new MEK is KEY-new:isklmsrv. For more information see “[Encryption keys](#)” on page 735.
2. **Encrypting new files:** This step describes how to update the policy rules that specify how files are encrypted so that new files are encrypted with FEKs that are wrapped with the new MEK:
 - a) Find the ENCRYPTION IS rules in your encryption policy.
 - b) In the KEYS parameter, replace the name of the old MEK with the name of the new MEK, as in the following example:

```
RULE 'EncRule1' ENCRYPTION 'E1' IS  
      ALGO 'DEFAULTNISTSP800131A'  
      KEYS('KEY-new:isklmsrv')
```

- c) Issue the **mmchpolicy** command to change the policy rules to encrypt new files with the new policy.

For more information, see “[Encryption policy rules](#)” on page 736.

3. **Rewrapping the FEKs of existing files:** This step describes how to rewrap the FEKs of existing files with the new MEK:

- a) Create a CHANGE ENCRYPTION KEYS policy rule to rewrap FEKs that are wrapped with the old MEK. This rule scans a specified group of files, unwraps each FEK entry that is wrapped with the old MEK, and rewraps the FEK entry with the new MEK. In the following example the rule finds all the files that are wrapped with KEY-old:isklmsrv and rewraps them with KEY-new:isklmsrv:

```
RULE 'Rule to rewrap keys' CHANGE ENCRYPTION KEYS FROM 'KEY-old:isklmsrv' to 'KEY-  
new:isklmsrv'
```

This rule has optional POOL, FILESET, SHOW, and WHERE clauses to specify the group of files to be rewrapped. For more information, see “[Encryption policy rules](#)” on page 736.

- b) Issue the **mmapplypolicy** command to apply the policy rule that you created in Step 3(a). The command rewraps the FEKs of the existing files with the new MEK.

Note: The first phase of the **mmapplypolicy** command's operation can be a lengthy process. In this phase the command scans all of the files in the affected file system or filesset to discover files that meet the criteria of the policy rule. If your file system or filesset is very large, you might want to delay issuing the **mmapplypolicy** command until a time when the system is not running a heavy load of applications. For more information see “[Phase one: Selecting candidate files](#)” on page 553.

Note: The **mmapplypolicy** command does not process files in unlinked filessets. If these files are encrypted and the FEKs are wrapped with the old MEK, and if the old MEK is deleted from the RKM server, the data in these files is unrecoverable.

4. Delete any snapshots that might contain files that are encrypted with the old MEK (KEY-old:isklmsrv).



Warning: You will not be able to delete such snapshots after the old MEK is deleted from the key server.

Do not begin the next step until the **mmapplypolicy** command from Step 3(b) has completed.

5. If the old MEK is no longer needed, delete it from the RKM server. In the regular encryption setup, open the RKM server console and delete the old MEK. In the simplified encryption setup, issue the **mmkeyserv key delete** command to delete the MEK.

Note: When you delete a MEK from the RKM server, any file that is encrypted with an FEK that is still wrapped by the old MEK cannot be decrypted and its data is unrecoverable.

- Delete the old MEK from the key cache on each node. The old MEK is present in the key cache of any node that did I/O operations to a file whose FEK was wrapped with the old MEK. To delete the old MEK, issue the following command on each node where the old MEK is cached:

```
/usr/lpp/mmfs/bin/tsctl encKeyCachePurge 'KEY-old:isklmsrv'
```

For more information see the subtopic "Secure deletion and encryption key cache purging" in the help topic ["Secure deletion" on page 856](#).



Warning: If the steps for key rotation are not followed carefully, they can result in unrecoverable data loss. Be aware of the following issues:

- Check other file systems that might contain files that are encrypted with the old MEK. If there are such files, rewrap their FEKs with the new MEK before you delete the old MEK from the RKM server.
- Test the policy rule by running the **mmapplypolicy** command with the **-I test** option. Check the output to verify that the policy rule is selecting the correct set of files. Also verify that the CHANGE ENCRYPTION KEYS statement specifies the correct old MEK and new MEK.
- To preserve the data in files that were deleted or were unlinked from filesets, restore the files (from a backup or snapshot, if available) before you issue the **mmapplypolicy** command. Remember that the **mmapplypolicy** command does not process unlinked files that were deleted from filesets with operating system commands such as **rm** and **unlink**.

Encryption and standards compliance

IBM Storage Scale encryption enables the use of FIPS 140-2-certified cryptography and also complies with the recommendations of NIST SP800-131A.

Encryption and FIPS 140-2 certification

The **FIPS1402mode** attribute of the **mmchconfig** command enables or disables the use of FIPS 140-2 certified cryptography in encrypted communications between nodes and in file encryption.

Important: IBM Storage Scale uses IBM Global Security Kit (GSKit) as the underlying cryptographic engine. In July 2022, the GSKit FIPS 140-2 certificate status was changed to *historical*.

The **FIPS1402mode** attribute controls whether the use of crypto-based security mechanisms (if they are to be used at all, per the IBM Storage Scale administrator) is to be provided by software modules that are certified according to the requirements and standards that are described by the Federal Information Processing Standards (FIPS) 140 Publication Series. When in FIPS 140-2 mode, IBM Storage Scale uses the FIPS 140-2 approved cryptographic provider IBM Crypto for C (ICC) (certificate 3064) for cryptography. The certificate is listed on the NIST website.

To enable FIPS 140-2 mode, issue the following command:

```
mmchconfig FIPS1402mode=yes
```

To disable FIPS 140-2 mode, issue the following command:

```
mmchconfig FIPS1402mode=no
```

When it is enabled, FIPS 140-2 mode applies only to the following two features of IBM Storage Scale:

- Encryption and decryption of file data when it is transmitted between nodes in the current cluster or between a node in the current cluster and a node in another cluster. To enable this feature, issue the following command:

```
mmchconfig cipherList=SupportedCipher
```

where *SupportedCipher* is a cipher that is supported by IBM Storage Scale, such as AES128-GCM-SHA256. For more information, see the following topics:

- *Security mode in the IBM Storage Scale: Administration Guide*.
- *Setting security mode for internode communications in a cluster in the IBM Storage Scale: Administration Guide*.
- Encryption of file data as it is written to storage media and decryption of file data as it is read from storage media. For more information about file data encryption, see the following section of the documentation:
 - *Encryption in the IBM Storage Scale: Administration Guide*.

Note: For performance reasons, do not enable FIPS 140-2 mode unless all the nodes in the cluster are running FIPS-certified kernels in FIPS mode. This note applies only to encryption of file data as it is written to storage media and decryption of file data as it is read from storage media. This note does not apply to encryption and decryption of file data when it is transmitted between nodes.

FIPS 140-2 mode does not apply to other components of IBM Storage Scale that use encryption, such as object encryption.

For more information, see the topic *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Limitation in IBM Storage Scale 4.2.0 and earlier with POWER8, little-endian

In IBM Storage Scale 4.2.0 and earlier, in a POWER8, little-endian environment, the setting **FIPS1402mode=no** is required for the following operations:

- File encryption
- Secure communications between nodes. For more information, see the following descriptions in the *IBM Storage Scale: Command and Programming Reference*:
 - The **-1 Cipherlist** parameter of the **mmauth** command
 - The **cipherList** parameter of the **mmchconfig** command
- CCR enablement. For more information, see the following descriptions in the *IBM Storage Scale: Command and Programming Reference*:
 - The **--ccr-enable** parameter of the **mmchcluster** command
 - The **--ccr-enable** parameter of the **mmcrccluster** command.

Encryption and NIST SP800-131A compliance

Encryption uses NIST-compliant mechanisms.

The mechanisms that are used by file encryption, including ciphers and key lengths, are compliant with the NIST SP800-131A recommendations. See *NIST Special Publication 800-131A, Revision 1* at <http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar1.pdf>.

Encryption in a multicluster environment

If an encrypted file system is made available via remote mounts, then the remote cluster also requires network reachability to the key server. In some deployments, the key servers might be located at the home cluster, but in others one might choose to locate key servers in a high-availability configuration on both home and remote clusters. The order of the servers can be specified in the set of RKM back ends such that the server closest to the local node is accessed first.

For more information, see the following topics:

- “Simplified setup: Accessing a remote file system” on page 774
- “Regular setup: Accessing a remote file system” on page 807

Encryption in a Disaster Recovery environment

While setting up multiple key servers in a high-availability configuration is important to ensure that the MEKs remain available, it is especially important in a Disaster Recovery environment. It is a good practice to place at least one key server on each site to ensure that keys remain available if access to an entire site is lost. For more information, see “[Adding backup RKM servers in a high-availability configuration](#)” on page 746.

Encryption and backup/restore

GPFS will deliver all data to mmbackup and other external backup solutions in **cleartext** whether or not the data is encrypted in GPFS. Any backups that are taken **will not** preserve the encryption status or the encrypted content of the data. Files that are recreated upon restore will be considered for encryption status based on the policy in place on the file system at the time of the restore operation.

Encryption and snapshots

IBM Storage Scale preserves the encryption status of files when they are copied into global or fileset snapshots.

The global snapshot restore operation restores encrypted files and their FEKs and MEKs. For more information, see the topic *mmrestorefs command* in the *IBM Storage Scale Command and Programming Reference*.

As snapshots are taken of a file system or fileset that includes encrypted files, subsequent operations on the active files and snapshots depend on the continuing availability of the MEKs for those files.

Over time, some MEKs might no longer be accessible. For example, MEKs can be deleted from the server as a result of secure deletion. Similarly, encrypted files might be moved to a different key server and have their FEKs rewrapped with MEKs from the new server, possibly resulting in the old server being decommissioned.

All snapshots that include encrypted files whose MEKs will no longer be accessible must be deleted with the **mmdeletesnapshot** command before the current MEKs become unavailable. Otherwise, the corresponding snapshots will no longer be able to be removed, as is the case of the active files whose keys are no longer available.

Encryption and a local read-only cache (LROC) device

IBM Storage Scale holds encrypted file data in memory as cleartext. To support this design, IBM Storage Scale decrypts encrypted file data as it is read into memory and encrypts file data as it is written into an encrypted file.

By default, IBM Storage Scale does not allow cleartext from encrypted files to be copied into an LROC device. The reason is that a security exposure arises when cleartext from an encrypted file is copied into an LROC device. Because LROC device storage is non-volatile, an attacker can capture the cleartext by removing the LROC device from the system and reading the cleartext at some other location.

To enable cleartext from an encrypted file to be copied into an LROC device, you can issue the **mmchconfig** command with the attribute **LROCEnableStoringClearText=yes**. You might choose this option if you have configured your system in some way to remove the security exposure. One such method is to install an LROC device that internally encrypts data that is written into it and decrypts data that is read from it. But see the following warning.

Warning: If you allow cleartext from an encrypted file to be copied into an LROC device, you must take steps to protect the cleartext while it is in LROC storage. One method is to install an LROC storage device that internally encrypts data that is written into it and decrypts data that is read from it. However, be aware that a device of this type voids the IBM Storage Scale secure deletion guarantee, because IBM Storage Scale does not manage the encryption key for the device.

For more information, see the following links:

[Chapter 61, “Local read-only cache,” on page 975](#)

The topic *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Encryption and external pools

Encrypted files are migrated to external pools in cleartext and are re-encrypted when they are retrieved from external pools.

Whenever encrypted files on the IBM Storage Scale file system are migrated to an external storage pool, they are decrypted before migration to the external storage pool takes place. Files are sent to the tool that manages the external storage in cleartext, leaving file stubs in the file system. When these migrated files are recalled, they are retrieved in cleartext and are subsequently re-encrypted by IBM Storage Scale as they are rewritten to disk. Typically the product software that manages the external storage provides the means to encrypt the cleartext data sent by IBM Storage Scale before writing the data to the external storage. Similarly the product software can decrypt the data before sending it to IBM Storage Scale when the file is recalled.

When the stub files that are created from the migration of data to an external pool are copied to other locations in the file system, IBM Storage Scale recalls the data from the external pool if the destination of the copy is a different file (inode) space. For example, copying a stub file from one file system to another or from one independent fileset to another triggers the recall of the file data from the external pool. If the placement policy for the destination of the file copy requires files to be encrypted, then the file also is encrypted when recalled.

For more information about external pools, see [“External storage pools” on page 530](#).

Encryption requirements and limitations

Learn the requirements and limitations for using file encryption.

For encryption requirements, see the topic [“Preparation for encryption” on page 742](#).

Encryption has the following requirements and limitations:

- Existing files cannot be encrypted. To encrypt a file that is currently not encrypted, you must copy it into a new file whose encryption policy rules dictate that the file is to be encrypted. Note that renaming a file does not change its encryption attributes. Encryption attributes are defined at the time that the file is created.
- The following types of nodes must have network connectivity to the key server node so that they can retrieve the master encryption key (MEK), which is needed to encrypt or decrypt file data:
 - The file system manager node.

Note: Bear in mind that the file system manager node can be changed. For more information, see the topic *mmchmgr command* and *mmlsmgr command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

- An NSD server node.
- Any node that might participate in a maintenance operation, such as restriping the file system.
- For a multicluster environment, see the topic [“Encryption in a multicluster environment” on page 861](#).
- For a Disaster Recovery environment, see the topic [“Encryption in a Disaster Recovery environment” on page 862](#).
- For backup and restore, see the topic [“Encryption and backup/restore” on page 862](#).
- For snapshots, see the topic [“Encryption and snapshots” on page 862](#).
- Data for encrypted files is not stored in the inode. For more information, see the topic *Use of disk storage and file structure within a GPFS file system* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

- Data from encrypted files is not stored in the highly available write cache (HAWC). For more information, see [Chapter 60, “Highly available write cache \(HAWC\),” on page 971](#).
- Encryption is not supported on Windows. The encryption function must remain disabled when Windows nodes are added to the cluster.
- To avoid a security exposure, by default IBM Storage Scale does not allow file data from encrypted files, which is held in memory as cleartext, to be copied into an LROC. As a result, a file system in which most of the files are encrypted does not take advantage of the performance benefits that are provided by an LROC. However, you can set IBM Storage Scale to enable cleartext from encrypted files to be copied into an LROC. You might choose this option if you can configure your system to remove the security problem.

Warning: If you allow cleartext from an encrypted file to be copied into an LROC, you must take steps to protect the cleartext while it is in LROC storage.

For more information, see the following links:

[“Encryption and a local read-only cache \(LROC\) device” on page 862](#)

[Chapter 61, “Local read-only cache,” on page 975](#)

The topic *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Chapter 50. Managing certificates to secure communications between GUI web server and web browsers

The IBM Storage Scale system supports self-signed and trusted certificates that are provided by a certificate authority (CA) to secure communications between the system and web browser.

During system setup, an initial self-signed certificate is created to use for secure connections between the GUI web servers and web browsers. Based on the security requirements for your system, you can create either a new self-signed certificate or install a signed certificate that is created by the certifying authority. Self-signed certificates can generate web browser security warnings and might not comply with organizational security guidelines.

The trusted certificates are created by a third-party certificate authority. These certificate authorities ensure that certificates have the required security level for an organization based on purchase agreements. Trusted certificates usually have higher security controls for encryption of data and do not cause browser security warnings. Trusted certificates are also stored in the Liberty profile SSL keystore.

Major web browsers trust the CA-certified certificates by default. Hence, they can confirm that the certificate that is received by the GUI server can be trusted. You can either buy a signed certificate from a trusted third-party authority or create your own certificate and get it certified. You can use both self-signed and trusted certificates. However, using a trusted certificate is the preferred way because the browser trusts this certificate automatically without any manual interventions.

You can either use the **Services > GUI** page in the GUI or CLI to install and use the certificates.

Obtain and import certificates by using the GUI

You can use the **Services > GUI** page in the GUI to complete the following tasks:

1. Generate a self-signed certificate by using the **Install Self-Signed Certificate** option.
2. Generate a certificate request and install it after getting it certified by the CA by using the **Create Certificate Request** option.

Note: You can use new attributes for *Subject Alternative Names*, if the OpenSSL version on the GUI node is 1.1.1 or later.

3. Install an already issued certificate by using the **Import Certificate** option.
4. View the details of the certificate that is applied on the local GUI node by using the **View Certificate** option.

Obtain and import a signed-certificate from a trusted certificate authority by using CLI

You need to complete the following steps to obtain and import a signed-certificate from a trusted certificate authority:

1. Generate a private key by issuing the following command:

```
openssl genrsa -out <nameOfYourKey>.key 2048
```

2. Generate the certificate request as shown in the following example:

```
openssl req -new -key <nameOfYourKey>.key -out <nameOfYourKey>.csr
```

The system prompts you to enter the following details:

```
Country Name (2 letter code) [XX]:  
State or Province Name (full name) []:  
Locality Name (eg, city) [Default City]:  
Organization Name (eg, company) [Default Company Ltd]:  
Organizational Unit Name (eg, section) []:  
Common Name (eg, your name or your server's hostname) []:  
Email Address []:  
Please enter the following 'extra' attributes to be sent with your certificate request  
A challenge password []:  
An optional company name []:
```

3. Send the certificate request to a trusted certificate authority to get a certificate file.
4. Create a PKCS12 store that contains the certificate as shown in the following example:

```
openssl pkcs12 -export -in <yourCertificateFile> -inkey <nameOfYourKey>.key -out  
<nameOfYourPKCS12File>.p12
```

The system prompts to set the export password as shown in the following example:

```
Enter export Password: <yourPassword>  
Verifying - Enter export Password: <yourPassword>
```

5. Generate a Java keystore file (.jks) by using the keytool. Issue the following commands to generate the file.

```
/usr/lpp/mmfs/java/jre/bin/keytool -importkeystore -srckeystore  
<NameOfYourPKCS12File>.p12 -destkeystore  
<NameOfYourJKSFile>.jks -srcstoretype pkcs12
```

The system prompts you to enter the destination keystore password. You need to use the same password that you used when you created the PKCS12 store.

```
Enter destination keystore password: <yourPassword>  
Re-enter new password: <yourPassword>  
Enter source keystore password: <yourPassword>
```

6. If you want to encode your password in XOR so that it does not get stored in plain text, use a security utility as shown in the following example:

```
/opt/ibm/wlp/bin/securityUtility encode <yourPassword>
```

7. Issue the following command:

```
/usr/lpp/mmfs/gui/cli/sethttpskeystore <pathToKeystore>.jks
```

This command imports the keystore into the WebSphere configuration, which can be used for secure connections. You are prompted to insert your keystore password. You can use either plain text or the XOR password, which you created in the previous step.

Note: The command /usr/lpp/mmfs/gui/cli/lshttpskeystore shows an active custom keystore with a user-defined certificate. If you want to return to the default GUI certificate issue /usr/lpp/mmfs/gui/cli/rmhttpskeystore.

Chapter 51. Securing protocol data

The data cannot be secured only by authenticating and authorizing the users to access the data. You also need to ensure that the communication channel that is used to raise authentication requests and data transfer is secured. The security features associated with the protocols that you use to store and access data also help to provide data in transit security for the protocol data.

The secured data access by clients through protocols is achieved through the following two steps:

1. Establishing secured connection between the IBM Storage Scale system and the authentication server.

When the client raises an authentication request to access the data, the IBM Storage Scale system interacts with the external authentication servers like Active Directory or LDAP based on the authentication configuration. You can configure the security services like TLS and Kerberos with the external authentication server to secure the communication channel between the IBM Storage Scale system and the external authentication server.

2. Securing the data transfer.

The actual data access wherein the data transfer is made secured with the security features that are available with the protocol that you use to access the data.

The following diagram depicts the data in transit security implementation in the IBM Storage Scale system.

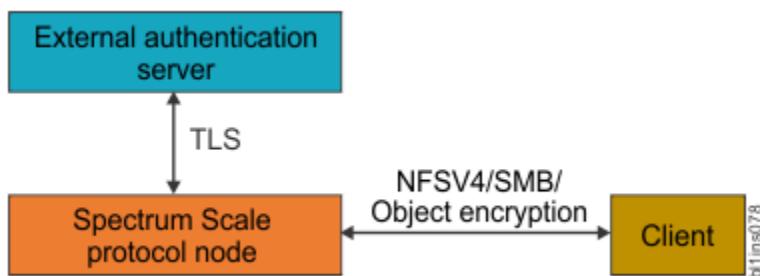


Figure 32. Implementation of data in transit security for protocol data

Secured connection between the IBM Storage Scale system and the authentication server

You can configure the following authentication servers to configure file and Swift Object access:

- Microsoft Active Directory (AD)
- Lightweight Directory Access Protocol (LDAP)
- Keystone

AD and LDAP can be used as the authentication server for both file and Swift Object access. Configuring the Keystone server is a mandatory requirement for the Swift Object access to function. The keystone needs to interact with the authentication server to resolve the authentication requests. You can configure either an internal or external keystone server for Swift Object access. The following table lists the security features that are used to secure the corresponding authentication server.

Table 67. Security features that are used to secure authentication server		
Authentication server	Supported protocols	Security features
Active Directory	File and Swift Object	Kerberos for file and TLS for Swift Object.
LDAP	File and Swift Object	Both TLS and Kerberos for file and only TLS for Swift Object.

Table 67. Security features that are used to secure authentication server (continued)		
Authentication server	Supported protocols	Security features
Keystone	Swift Object	SSL certificate to enable HTTPS connection

Secured data transfer

The secured data transfer over the network is based on the security features available with the protocols that are used to access the data.

Secured SMB data transfer

SMB protocol version 3 and later has the following capabilities to provide tighter security for the data transfers:

1. Secured dialect negotiation
2. Improved signing
3. Secured transmission

The dialect negotiation is used to identify the highest level dialect both server and client can support. The system administrator can enable SMB encryption by using the `server smb encrypt` setting at the export level. The following three modes are available for the secured SMB access:

- Automatic
- Mandatory
- Disabled

When the SMB services are enabled, the SMB encryption is enabled in the automatic mode by default.

Note: SMB supports per-export encryption, which allows the administrators to selectively enable or disable encryption per SMB share.

Secured NFS data transfer

The following security methods are used with NFSv4 protocol:

1. Enabling squashing

Any file requests that are made by the root user on the client system is considered as a potential threat. By default, root user requests are treated as if it is made by the user on the server. If you disable squashing, the root user on the client gets the same level of access to files on the system as the root user on the server. You can disable squashing if, for example, you want to run an administrative task on the client system that has the exported directories that are stored on it.

2. Using Kerberos

Kerberos is a network authentication protocol that ensures secure communication over a network. You can use Kerberos instead of local UNIX UIDs and GIDs to authenticate users. Kerberos can operate in the following modes to provide improved security:

- **Kerberos v5:** Authentication only
- **Kerberos v5 with integrity:** Authentication and data integrity
- **Kerberos v5 with privacy:** Authentication and encryption of data traffic between the client and the server. Most secure, but it might cause some performance issues because of the heavy processing required for encryption.

3. Enabling port security

You can enable or disable the port security in all communications between the client and the server. When port security is enabled, the system does not allow access to the requests that originate from ports where the port number is greater than the hardcoded threshold value of 1024. NFS port security is mapped to the "PRIVILEGEDPORT" configuration parameter. This option is available in the

mnnfs config command (for changing the system default) and in the **mnnfs export** command (for changing the individual exports). "PRIVILEGEDPORT" by default is set to "False", which means that there is no limit on the accepted source ports or that the port security is not enabled. To enable port security, enable the "PRIVILEGEDPORT" parameter with **mnnfs config change** or **mnnfs export create/change** command.

Secured S3 services for configuration data

By using the following executable and file modes you can enable encryption for a secret key:

executable

By default, the executable mode is set. The encryption keys are stored in the CCR.

```
# mmccr flist | grep _ces_s3
```

A sample output is as follows:

```
1 _ces_s3.master_keys
```

file

If you do not want to store an encryption key in the CCR, you can change the mode to **file** to store the key. The encryption key is stored in the CES shared root path.

```
# mms3 config change NC_MASTER_KEYS_STORE_TYPE=file
```

Planning for protocol data security

It is recommended to adhere to the following best practices when you plan to set up data security:

- When Windows clients use SMB 3.0, always configure SMB share with `server smb encrypt = mandatory`.
- Avoid clients who use SMB 2.1 from accessing SMB data.
- Always enforce to use UNC with NetBIOS name of the cluster to access SMB data.
- To ensure NFSV4 encryption, use an authentication method that is configured with Kerberos.
- For secure communication between the client system and IBM Storage Scale Object, the system administrator must configure HAProxy with Secure Sockets Layer (SSL) support.

Related concepts

[Data security limitations](#)

The following are the protocol data security limitations:

Related tasks

[Configuring protocol data security](#)

The data security features associated with protocols facilitate to configure a secured way for the clients to raise the data access request and to transfer data from the IBM Storage Scale system to the client system.

Configuring protocol data security

The data security features associated with protocols facilitate to configure a secured way for the clients to raise the data access request and to transfer data from the IBM Storage Scale system to the client system.

Related concepts

[Planning for protocol data security](#)

It is recommended to adhere to the following best practices when you plan to set up data security:

[Data security limitations](#)

The following are the protocol data security limitations:

Enabling secured connection between the IBM Storage Scale system and authentication server

You need to secure the communication channel between the IBM Storage Scale system and authentication server to secure the authentication server and hence to prevent unauthorized access to data and other system resources.

Securing AD server

To secure the AD server that is used for file access, configure it with Kerberos and to secure AD used for object access, configure it with TLS.

In the AD-based authentication for file access, Kerberos is configured by default. The following steps provide an example on how to configure TLS with AD, while it is used for object access.

1. Ensure that the CA certificate for AD server is placed under /var/mmfs/tmp directory with the name *object_ldap_cacert.pem*, specifically on the protocol node where the command is run. Perform validation of CA cert availability with desired name at required location as shown in the following example:

```
# stat /var/mmfs/tmp/object_ldap_cacert.pem
File: /var/mmfs/tmp/object_ldap_cacert.pem
Size: 2130 Blocks: 8 IO Block: 4096 regular file
Device: fd00h/64768d Inode: 103169903 Links: 1
Access: (0644/-rw-r--r--) Uid: ( 0/ root) Gid: ( 0/ root)
Context: unconfined_u:object_r:user_tmp_t:s0
Access: 2015-01-23 12:37:34.088837381 +0530
Modify: 2015-01-23 12:16:24.438837381 +0530
Change: 2015-01-23 12:16:24.438837381 +0530
```

2. To configure AD with TLS authentication for object access, issue the **mmuserauth service create** command:

```
# mmuserauth service create --type ad --data-access-method object
--user-name "cn=Administrator,cn=Users,dc=IBM,dc=local" --base-dn "dc=IBM,DC=local"
--enable-server-tls --ks-dns-name myKeystoneDnsName
--ks-admin-user admin --servers myADserver
--user-id-attrib cn --user-name-attrib sAMAccountName
--user-objectclass organizationalPerson --user-dn "cn=Users,dc=IBM,dc=local"
--ks-swift-user swift
Object configuration with LDAP (Active Directory) as identity
backend is completed successfully.
Object Authentication configuration completed successfully.
```

Note: The value that you specify for --servers must match the value in the TLS certificate. Otherwise the command fails.

3. To verify the authentication configuration, use the **mmuserauth service list** command as shown in the following example:

```
# mmuserauth service list
FILE access not configured
PARAMETERS          VALUES
-----
OBJECT access configuration: AD
PARAMETERS          VALUES
-----
ENABLE_ANONYMOUS_BIND    false
ENABLE_SERVER_TLS        true
ENABLE_KS_SSL           false
USER_NAME               cn=Administrator,cn=Users,dc=IBM,dc=local
SERVERS                 myADserver
BASE_DN                 dc=IBM,DC=local
USER_DN                 cn=users,dc=ibm,dc=local
USER_OBJECTCLASS         organizationalPerson
USER_NAME_ATTRIB         sAMAccountName
USER_ID_ATTRIB           cn
USER_MAIL_ATTRIB         mail
```

USER_FILTER	none
ENABLE_KS_CASIGNING	false
KS_ADMIN_USER	admin

Securing LDAP server

To secure the LDAP server that is used for file access, configure it with TLS and Kerberos and to secure LDAP server that is used for object access, configure it with TLS.

Provide examples of how to configure LDAP with TLS and Kerberos to secure the LDAP server when it is used for file and object access.

1. To configure LDAP with TLS and Kerberos as the authentication method for file access, issue the **mmuserauth service create** command as shown in the following example:

```
# mmuserauth service create --type ldap --data-access-method file
--servers es-pune-host-01 --base-dn dc=example,dc=com
--user-name cn=manager,dc=example,dc=com
--netbios-name ess --enable-server-tls --enable-kerberos
--kerberos-server es-pune-host-01 --kerberos-realm example.com
```

The system displays the following output:

```
File Authentication configuration completed successfully.
```

To verify the authentication configuration, use the **mmuserauth service list** command as shown in the following example:

```
# mmuserauth service list
```

The system displays the following output:

```
FILE access configuration : LDAP
PARAMETERS          VALUES
-----
ENABLE_ANONYMOUS_BIND    false
ENABLE_SERVER_TLS        true
ENABLE_KERBEROS         true
USER_NAME                cn=manager,dc=example,dc=com
SERVERS                  es-pune-host-01
NETBIOS_NAME              ess
BASE_DN                  dc=example,dc=com
USER_DN                  none
GROUP_DN                  none
NETGROUP_DN                none
USER_OBJECTCLASS          posixAccount
GROUP_OBJECTCLASS         posixGroup
USER_NAME_ATTRIB          cn
USER_ID_ATTRIB            uid
KERBEROS_SERVER           es-pune-host-01
KERBEROS_REALM            example.com

OBJECT access not configured
PARAMETERS          VALUES
-----
```

2. To configure LDAP with TLS as the authentication method for object access, issue the **mmuserauth service create** command as shown in the following example:

```
# mmuserauth service create --type ldap --data-access-method object
--user-name "cn=manager,dc=essldapdomain"
--base-dn dc=isst,dc=aus,dc=stglabs,dc=ibm,dc=com --enable-server-tls
--ks-dns-name c40bbc2xn3 --ks-admin-user mamdouh --servers 192.0.2.11
--user-dn "ou=People,dc=essldapdomain" --ks-swift-user swift
```

The system displays the following output:

```
Object configuration with LDAP as identity backend is completed successfully.
Object Authentication configuration completed successfully.
```

To verify the authentication configuration, use the **mmuserauth service list** command as shown in the following example:

```
# mmuserauth service list
```

The system displays the following output:

```
FILE access not configured
PARAMETERS          VALUES
-----
OBJECT access configuration : LDAP
PARAMETERS          VALUES
-----
ENABLE_ANONYMOUS_BIND    false
ENABLE_SERVER_TLS        true
ENABLE_KS_SSL            false
USER_NAME                cn=manager,dc=essldapdomain
SERVERS                  192.0.2.11
BASE_DN                   dc=isst,dc=aus,dc=stglabs,dc=ibm,dc=com
USER_DN                   ou=people,dc=essldapdomain
USER_OBJECTCLASS          posixAccount
USER_NAME_ATTRIB          cn
USER_ID_ATTRIB            uid
USER_MAIL_ATTRIB          mail
USER_FILTER                none
ENABLE_KS_CASIGNING       false
KS_ADMIN_USER              mAMDouH
```

Securing Keystone server

The Keystone server that is used by the IBM Storage Scale system supports SSL. The SSL certificate provides secure communication while resolving the authentication requests. When Keystone is configured with authentication servers such as LDAP or AD, the system can be configured to establish a secured communication between AD or LDAP and Keystone by using TLS encryption. For more information on configuring AD or LDAP-based authentication with TLS, see the **mmuserauth service create** command. The IBM Storage Scale for Object Storage can also be configured with an external Keystone server. If the external Keystone server contains SSL certificate in place, then the system administrator can configure secured communication with the IBM Storage Scale system by following some manual steps.

The following is an example on how to configure secured object access.

1. Remove the object authentication and the ID mapping:

```
/usr/lpp/mmfs/bin/mmuserauth service remove --data-access-method object
/usr/lpp/mmfs/bin/mmuserauth service remove --data-access-method object --idmapdelete
mmuserauth service list
```

The system displays the following output:

```
FILE access not configured
PARAMETERS          VALUES
-----
OBJECT access not configured
PARAMETERS          VALUES
-----
```

2. Copy the CA certificate on the node on which the **mmuserauth** command is being run. The name and the path of the CA certificate on the current node is /var/mmfs/tmp/ks_ext_cacert.pem.
3. Configure object authentication by using the **mmuserauth service create** command with the --enable-ks-ssl option:

```
mmuserauth service create --data-access-method object --enable-ks-ssl --type
userdefined --ks-ext-endpoint https://externalkeyonesteserver:35357/v3
--ks-swift-user swift
```

4. Run the **mmuserauth service list** command to verify the configuration:

```
mmuserauth service list
FILE access not configured
```

PARAMETERS	VALUES
OBJECT access configuration	: USERDEFINED
PARAMETERS	VALUES

Securing data transfer

The data in transit security is configured by using the security features that are available with the protocol that is used for data I/O.

Securing NFS data transfer

Securing the NFS data transfer over the network is achieved by using the Kerberos-based encryption that is available with NFSV4 protocol. You can use Kerberos to encrypt the data that is transferred over the network and also to secure the communication with the authentication server.

The following example shows how to enable data security to ensure secured NFS data transfer.

1. Create a keytab file for protocol nodes in IBM Storage Scale cluster. To create a keytab file, you need to create a principal nfs/<node-fqdn> for each protocol node. Issue the following commands on the system that hosts the KDC server. In the following example, the sample commands are submitted on the Linux system that hosts MIT KDC server:

```
$ addprinc -randkey nfs/<protocol-node1-fqdn>
$ addprinc -randkey nfs/<protocol-node2-fqdn>

.....
&mldr; $ addprinc -randkey nfs/<protocol-nodeN-fqdn>
$ ktadd -k /tmp/krb5.keytab nfs/<protocol-node1-fqdn>
$ ktadd -k /tmp/krb5.keytab nfs/<protocol-node2-fqdn>

.....
&mldr; $ ktadd -k /tmp/krb5.keytab nfs/<protocol-nodeN-fqdn>
```

2. Ensure that the keytab file that is created is placed under the /tmp directory as krb5_scale.keytab, specifically on the node where the IBM Storage Scale authentication commands are submitted. Perform validation of keytab file availability with the required name and location:

```
# stat /var/mmfs/tmp/krb5_scale.keytab
  File: /var/mmfs/tmp/krb5_scale.keytab
  Size: 1490        Blocks: 8          IO Block: 4096   regular file
Device: fd00h/64768d Inode: 68252098      Links: 1
Access: (0600/-rw-----)  Uid: (     0/    root)  Gid: (     0/    root)
Context: unconfined_u:object_r:user_tmp_t:s0
Access: 2021-05-26 06:52:49.511820164 -0400
Modify: 2021-04-28 09:52:07.661820164 -0400
Change: 2021-05-26 05:15:09.837820164 -0400
Birth: -
```

3. Issue the **mmuserauth service create** command on the IBM Storage Scale protocol node as shown in the following example:

```
# mmuserauth service create --data-access-method file --type ldap
--servers 192.0.2.17 --base-dn dc=example,dc=com
--user-name "cn=manager,dc=example,dc=com" --enable-kerberos
--kerberos-server 192.0.2.17 --kerberos-realm example.com --netbios-name cktest
File Authentication configuration completed successfully.
```

4. Issue the **mmuserauth service list** command to see the current authentication configuration as shown in the following example:

```
# mmuserauth service list
FILE access configuration : LDAP
PARAMETERS          VALUES
-----
```

```

ENABLE_ANONYMOUS_BIND      false
ENABLE_SERVER_TLS          false
ENABLE_KERBEROS            true
USER_NAME                  cn=manager,dc=example,dc=com
SERVERS                    9.118.46.17
NETBIOS_NAME               ckttest
BASE_DN                     dc=example,dc=com
USER_DN                     none
GROUP_DN                   none
NETGROUP_DN                none
USER_OBJECTCLASS           posixAccount
GROUP_OBJECTCLASS          posixGroup
USER_NAME_ATTRIB           cn
USER_ID_ATTRIB              uid
KERBEROS_SERVER             9.118.46.17
KERBEROS_REALM              example.com

OBJECT access not configured
PARAMETERS                 VALUES
-----
```

5. Create Kerberos exports with krb5, krb5i, and krb5p security features on the IBM Storage Scale node.

```

# mmcrfileset gpfs0 krb5
Fileset krb5 created with id 2 root inode 47898.

# mmlinkfileset gpfs0 krb5 -J /ibm/gpfs0/krb5
Fileset krb5 linked at /ibm/gpfs0/krb5

# mn nfs export add /ibm/gpfs0/krb5 --client \
  "*(ACCESS_TYPE=RW,SQUASH=no_root_squash,SECTYPE=krb5)"
The NFS export was created successfully.

# mmcrfileset gpfs0 krb5i
Fileset krb5i created with id 3 root inode 47900.

# mmlinkfileset gpfs0 krb5i -J /ibm/gpfs0/krb5i
Fileset krb5i linked at /ibm/gpfs0/krb5i

# mn nfs export add /ibm/gpfs0/krb5i --client \
  "*(ACCESS_TYPE=RW,SQUASH=no_root_squash,SECTYPE=krb5i)"
The NFS export was created successfully.

# mmcrfileset gpfs0 krb5p
Fileset krb5p created with id 4 root inode 47895.

# mmlinkfileset gpfs0 krb5p -J /ibm/gpfs0/krb5p
Fileset krb5p linked at /ibm/gpfs0/krb5p

# mn nfs export add /ibm/gpfs0/krb5p --client \
  "*(ACCESS_TYPE=RW,SQUASH=no_root_squash,SECTYPE=krb5p)"
The NFS export was created successfully.

# mn nfs export list
```

The system displays output similar to this:

Path	Delegations	Clients
/ibm/gpfs0/krb5	none	*
/ibm/gpfs0/krb5i	none	*
/ibm/gpfs0/krb5p	none	*
/ibm/gpfs0/nfsexp1	none	*

6. Issue the **mn nfs export list** command with krb5 option to see the authentication only configuration.

```
# mn nfs export list --nfsdefs /ibm/gpfs0/krb5
```

The system displays output similar to this:

Path	Delegations	Clients	Access_Type	Protocols	Transports	Squash	Anonymous_uid	Anonymous_gid	SecType	PrivilegedPort	Export_id	DefaultDelegation
/ibm/gpfs0/krb5	none	*	RW	3,4	TCP	NO_ROOT_SQUASH -2	-2	KRB5	FALSE		2	

7. Issue the **mmnfs export list** command with krb5i option to see the authentication and data integrity configuration.

```
# mmnfs export list --nfsdefs /ibm/gpfs0/krb5i
```

The system displays output similar to this:

Path	Delegations	Clients	Access_Type	Protocols	Transports	Squash	Anonymous_uid	Anonymous_gid	SecType	PrivilegedPort	Export_id
DefaultDelegation	Manage_Gids	NFS_Commit									
/ibm/gpfs0/krb5i	none	none	*	RW	3,4	TCP	NO_ROOT_SQUASH	-2	-2	KRB5I	FALSE
none	none	FALSE	FALSE								3

8. Issue the **mmnfs export list** command with krb5p option to see the authentication and privacy configuration.

```
# mmnfs export list --nfsdefs /ibm/gpfs0/krb5p
```

The system displays output similar to this:

Path	Delegations	Clients	Access_Type	Protocols	Transports	Squash	Anonymous_uid	Anonymous_gid	SecType	PrivilegedPort	Export_id
DefaultDelegation	Manage_Gids	NFS_Commit									
/ibm/gpfs0/krb5p	none	none	*	RW	3,4	TCP	NO_ROOT_SQUASH	-2	-2	KRB5P	FALSE
none	none	FALSE	FALSE								4

Securing SMB data transfer

Secured SMB data transfer can be enabled when you are using SMB3 and later.

You can either enable or disable encryption of the data in transit by using the **mmsmb export add** command as shown in the following example:

```
# mmsmb export add secured_export /ibm/gpfs0/secured_export --option "server smb encrypt=mandatory"
```

Secured object data transfer

For secure communication between the clients and the IBM Storage Scale Object, the system administrator needs to configure HAProxy for SSL termination, traffic encryption, and load balancing of the requests to IBM Storage Scale Object. The HAProxy needs to be set up on an external system that is not a part of the IBM Storage Scale cluster. For more information on how to configure HAProxy, see the documentation of the corresponding Linux distribution that you selected.

Data security limitations

The following are the protocol data security limitations:

- The SMB encryption is available only on SMB3 and later. All the limitations that are identified by Microsoft also apply to SMB encryption. There are no SMB encryption limitations that are specific to IBM Storage Scale.
- Delegations cannot be used with NFS in the Kerberos environment, because they cause the NFSV4 server to crash. If you use NFS in the Kerberos environment, you should disable delegations.

Related concepts

[Planning for protocol data security](#)

It is recommended to adhere to the following best practices when you plan to set up data security:

Related tasks

[Configuring protocol data security](#)

The data security features associated with protocols facilitate to configure a secured way for the clients to raise the data access request and to transfer data from the IBM Storage Scale system to the client system.

Chapter 52. Cloud services: Transparent cloud tiering and Cloud data sharing

This topic provides a brief description about managing your cloud storage using IBM Storage Scale.

Administering files for Transparent cloud tiering

You can administer files on the cloud storage account when you use Transparent cloud tiering.

When you use Transparent cloud tiering, there are maintenance tasks that must be done. It is recommended that you put them in a scheduler to make sure that the maintenance activity is performed. Since these maintenance activities affect overall data throughput, schedule them one at time (do not schedule them simultaneously) during non-peak demand times.

1. Reconcile your files once a month to make sure that the cloud directories that are maintained by Transparent cloud tiering services and the file system are synchronized.
2. Do a full backup of the cloud directory once a month to allow for faster and cleaner handling of disaster recovery and service problems.
3. Run the cloud destroy utility to remove files that are deleted from the file system from the cloud system.

These steps must be run for each Transparent cloud tiering container in the file system.

Note: You do not have to perform these actions for inactive containers that are not being migrated to (and have no delete activity).

See the information below for detailed instructions on how to perform these maintenance steps.

Applying a policy on a Transparent cloud tiering node

This topic provides description with an example about creating an ILM policy for tiering and then applying this policy to a Transparent cloud tiering node.

After a cloud account is configured, you can apply an ILM policy file to configure a cloud storage tier. The policy configuration is done by using IBM Storage Scale standard ILM policy query language statements.

For more information on ILM policies, see [Chapter 39, “Information lifecycle management for IBM Storage Scale,” on page 525](#).

You must create a policy and then apply this policy on the Cloud services node for the ILM-based migration and recall to work for the cloud storage tier.

Note: Administrators must consider appropriate high and low disk utilization threshold values that are applicable in the data center environment.

A sample policy rule and the steps to apply the policy on a node are as follows:

```
/* Sample policy.rules file for using Gateway functionality */
/* Define an external pool for the off-line storage */
define(
exclude_list,
(
  FALSE
  OR PATH_NAME LIKE '%/.mcstore/%'
)
)
define(
access_age,
(DAYS(CURRENT_TIMESTAMP) - DAYS(ACCESS_TIME))
)
define(
mb_allocated,
```

```

(INTEGER(KB_ALLOCATED / 1024))
)
define(
weight_expression,
(CASE
/*== The file is very young, the ranking is very low ===*/
WHEN access_age <= 1 THEN 0
/*== The file is very small, the ranking is low ===*/
WHEN mb_allocated < 1 THEN access_age
/*== The file is resident and large and old enough,
the ranking is standard ===*/
ELSE mb_allocated * access_age
END)
)
/* Define an external pool for the off-line storage */
RULE EXTERNAL POOL 'mcstore' EXEC '/opt/ibm/MCStore/bin/mcstore' OPTS '-F'
/* Define migration rule with a threshold to trigger low space events
and move data to the external off-line pool. When on-line usage
exceeds 25% utilization, it will move the coldest files to off-line storage
until the on-line usage is reduced to 20% utilization level. Only files that have
data on-line are eligible for migration. */
RULE 'MoveOffline' MIGRATE FROM POOL 'system'
THRESHOLD(25,20)
WEIGHT(weight_expression)
TO POOL 'mcstore'
WHERE(KB_ALLOCATED > 0) AND NOT(exclude_list)
/* Define default placement rule */
RULE 'Placement' SET POOL 'system'

```

For more information on how to work with the external storage pools and related policies, see [“Working with external storage pools” on page 570](#).

Note: Ensure that only a single instance of the policy is applied to migrate data to the external cloud storage pool. This avoids any potential locking issues that might arise due to multiple policy instances that try to work on the same set of files.

To ensure proper invocation of the policy on reaching threshold limits, see [Threshold based migration using callbacks example](#).

In the sample policy, the ‘OpenRead’ & ‘OpenWrite’ rule sections represent the transparent recall of a migrated or non-resident file. Transparent cloud tiering software adds its own extended attributes (dmapi.MCEA) to each file it processes. Displacement 5 in the extended attributes indicate the resident state of the file. If it is ‘N’ (non-resident), the policy issues a recall request to bring back the data from the cloud storage to the local file system for the requested Read or Write operation.

To apply a threshold policy to a file system, see [“Using thresholds to migrate data between pools” on page 567](#).

IBM Storage Scale also gives administrators a way to define policies to identify the files for migration, and apply those policies immediately using the **mmapplypolicy** command. This is different from the threshold-based policies (which are applied by using the **mmchpolicy** command). The Transparent cloud tiering service currently does not support parallelism in migrating files simultaneously, but parallelism in the **mmapplypolicy** command can be used to improve the overall throughput. Additionally, parallelism can be achieved by using an ILM policy to migrate data or by driving separate, parallel CLI commands.

A sample command to apply a policy is given here:

```
mmapplypolicy gpfs0 -P <rules.file> -m 24 -B 100 -g <global-work-directory> -N <tct-nodeclass>
```

where,

- *gpfs0* indicates the IBM Storage Scale system
- *-m* indicates the number of threads created and dispatched during policy execution phase. Use the **mmcloudgateway** command configuration tuning settings to set your migrate or recall thread counts.

Note: You must know the number of processors that are available on your Transparent cloud tiering service node.

- *-B* indicates the maximum number of files passed to each invocation of the EXEC script specified in the *<rules.file>*

- `-g` indicates a global work directory where IBM Storage Scale ILM policy keeps temporary data. This location/folder should be outside the folder/location being migrated. Otherwise, any temporary files that policy generates might get picked up for migration too, and migration of those temporary files might fail if those are removed by policy while they are being migrated.
- `-N` indicates the Transparent cloud tiering node class/nodes to which the migration workload would be distributed to further improve parallelism and in turn performance.

Note: These two parameters (`-m` and `-B`) can be adjusted to improve the performance of large-scale migrations.

The following sample policies are available in the package in the `/opt/ibm/MCStore/samples` folder:

Table 68. Sample policy list		
N o	Policy Name	Description
1	cloudDestroy.policy.template	Apply this policy for manually destroying orphaned cloud objects before retention time expires.
2	coresidentMigrate.template	Apply this policy for migrating files in the co-resident state, so that applications do not need to frequently recall files.
3	coResidenttoResident.template	Apply this policy if you want to convert all "co-resident" files in a file system to "resident".
	CoresToNonres.sobar.template	This is used during SOBAR restore to update the extended attributes (EAs) of co-resident files to non-resident, so that we could recall them on SOBAR restored site. Not required to be used outside SOBAR.
	exportfiles.policy.template	This is used to show how to export files from a given path. similar to migrateFromDirectory.policy.template. Can be used by customers.
4	listMigratedFiles.template	This policy will list all co-resident and resident files in the file system.
5	migrateFromDirectory.policy.template	This policy migrates all files in a specified directory to the cloud storage tier.
	migrateToSpecificCloudService.policy.template	This is used to show how to use a particular cloud service, to migrate files via policy to a particular cloud tier. Can be used by customers.
6	recallFromCloud.policy.template	Apply this policy to recall files from the cloud storage tier.
7	thresholdBasedMigration.policy.template	Apply this policy to automatically migrate files from the file system to the cloud storage upon reaching certain threshold levels.
8	thumbnailTransparentRecall.policy.template	This policy will help you display the thumbnails when files are listed in tools such as Windows Explorer.
9	transparentRecall.policy.template	Transparent recall pulls files from the cloud when they are accessed by an application (read or write).

Migrating files to the cloud storage tier

This topic provides a brief description on how to migrate files to the cloud storage tier by using Transparent cloud tiering.

Note: Before you try to migrate files to the cloud storage tier, ensure that your cloud service configuration is completed as summarized in [Chapter 7, “Configuring and tuning your system for Cloud services,” on page 87](#).

You can trigger migration of files from your file system to an external cloud storage tier either transparently or manually. Transparent migration is based on the policies that are applied on a file system. Data is automatically moved from the system pool to the configured external storage tier when the system pool reaches a certain threshold level. A file can be automatically migrated to or from cloud storage pool based on some characteristics of the file such as age, size, last access time, path. Alternatively, the user can manually migrate specific files or file sets to a cloud storage pool. For more information on policy-based migration, see [“Applying a policy on a Transparent cloud tiering node” on page 877](#).

To manually trigger migration of the file *file1*, issue this command:

```
mmcloudgateway files migrate file1
```

The state of the file becomes **Non-resident** after it is successfully migrated to the cloud storage tier.

If you want to migrate files in the co-resident state where the file has been copied to the cloud but also allows the data to be retained on the file system, see [“Pre-migrating files to the cloud storage tier” on page 880](#).

command. For more information on manually migrating files to the cloud storage tier, see **mmcloudgateway** command in the *IBM Storage Scale: Command and Programming Reference Guide*.

Pre-migrating files to the cloud storage tier

Using Transparent cloud tiering, you can migrate files in both co-resident and non-resident states depending on the type of data (warm/cold).

Normally, when a file is migrated to the cloud storage tier, the status of the file becomes non-resident. This means that, you need to completely recall the file from the cloud to be able to perform any read or write operation. This might be an issue when the data is warm. The calling application must recall the file every time it needs to perform any operation on the file, and this can be resource-intensive. The solution to this issue is, migrating files in co-resident state or pre-migration.

In this type of migration, irrespective of the file size, when the files are warm, they are archived to the cloud storage in the co-resident state. That allows applications to have continued access to the files without issuing any recall commands, but at the same time, ensures that data is at least available on the cloud if there is any type of disaster. As data gets colder, the files are migrated in the non-resident state. Since files are available both in the file system and on the cloud, the storage utilization is more here.

Small files whose data resides entirely in the inodes are migrated in the co-resident state even if they are cold.

Note: When files are pushed through NFS that need to be moved immediately to the cloud tier, be aware that CES NFS keeps NFSv3 files open and keeps them open indefinitely for performance reasons. Any files that are cached in this manner will not be migrated to by Transparent cloud tiering to the cloud tier. NFSv4 client caching is more measured and less likely to prevent files from being migrated to the cloud tier and is recommended for this sort of usage.

You can migrate files in the co-resident state by using a policy as well as the CLI.

You can use the following command to migrate files in the co-resident state, where the --co-resident-state keyword is mandatory:

```
mmcloudgateway files migrate --co-resident-state file1
```

To verify that the file is migrated in the co-resident state, issue the following command:

```
mmcloudgateway files list file1
```

The system displays an output similar to this:

```
File name      : /gpfs0/file1
On-line size  : 46
Used blocks   : 0
Data Version  : 1
Meta Version  : 1
State          : Co-resident
Base Name     : 57FA294111831B2B.10D9F57158C1628B.0063C1580F522F09.0000000000000000.5713748E.0000000000009E2B
```

For transparent migration in the co-resident state, the following policy has to be applied to the files by using the **mmchpolicy** command. A sample policy is available here: /opt/ibm/MCStore/samples/coresidentMigrate.template:

```
***** Licensed Materials - Property of IBM ****
* OCO Source Materials
* (C) Copyright IBM Corp. 2017 All Rights Reserved
* The source code for this program is not published or otherwise
* wise divested of its trade secrets, irrespective of what has
* been deposited with the U.S. Copyright Office.
*****
define(
    exclude_list,
    (
        FALSE
        OR PATH_NAME LIKE '%/.mcstore/%'
        OR PATH_NAME LIKE '%/.mcstore.bak/%'
    )
)

/* Define premigrate pool, where files are migrated in co-resident state. This represent files moved
to cloud but also available locally on Scale file system.
* It is to be used for warmer data, as that data needs to be available locally on Scale file system
too, to avoid cloud round trips.
*/
RULE EXTERNAL POOL 'premigrate' EXEC '/usr/lpp/mmfs/bin/mmcloudgateway files' OPTS '--co-resident-state -F'

/* Define migrate pool, where files are migrated in non-resident state. This represent files are moved
to cloud and are not available locally.
* It is to be used for colder data depending on file size. Larger colder files are made non-resident,
where as smaller files (less than 4K) are kept co-resident.
*/
RULE EXTERNAL POOL 'migrate' EXEC '/usr/lpp/mmfs/bin/mmcloudgateway files' OPTS '-F'

/* This rule defines movement of warm data. Each file (irrespective of it's size) is moved to cloud
in a co-resident state.
* It means, file is available on the cloud and, access to it is possible from the hot-standby site
if needed.
* Here the sample time interval to indicate warm data is, data that is not accessed between 10
to 30 days.
* We don't want to pick up HOT data that is being accessed in last 10 days.
* Another advantage of this co-resident migration is when data eventually gets colder, since it
is already migrated to cloud, only file truncation happens later.
*/
RULE 'MoveWarmData' MIGRATE FROM POOL 'system'
    THRESHOLD(0,0)
    TO POOL 'premigrate'
        WHERE NOT(exclude_list) AND
            (CURRENT_TIMESTAMP - ACCESS_TIME > INTERVAL '10' DAYS) AND
            (CURRENT_TIMESTAMP - ACCESS_TIME < INTERVAL '30' DAYS)

/* This rule defines movement of large files that are cold. Here, files that are above 4KB in size
are made non-resident to save
* space on Scale file system. For files that are smaller than 4KB are anyway stored in inode
block itself.
*/
RULE 'MoveLargeColdData' MIGRATE FROM POOL 'system'
    THRESHOLD(0,0)
    TO POOL 'migrate'
        WHERE(KB_ALLOCATED > 4) AND NOT(exclude_list) AND
            (CURRENT_TIMESTAMP - ACCESS_TIME > INTERVAL '30' DAYS)

/* This rule defines movement of smaller files that are cold. Here, files that are less than
4KB in size are made co-resident, as
* there is no saving in moving these files, as data resides within the inode block, and not
on disk. It avoids un-necessary recall cycles.
*/
RULE 'MoveSmallColdData' MIGRATE FROM POOL 'system'
    THRESHOLD(0,0)
    TO POOL 'premigrate'
        WHERE(KB_ALLOCATED < 4) AND NOT(exclude_list) AND
            (CURRENT_TIMESTAMP - ACCESS_TIME > INTERVAL '30' DAYS)

/* Define default placement rule */
RULE 'Placement' SET POOL 'system'
```

Recalling files from the cloud storage tier

This topic provides a brief description on how to recall files from the cloud storage tier by using Transparent cloud tiering.

You can trigger recall of files from the cloud storage tier either transparently or manually. Transparent recall is based on the policies that are applied on a file system. Data is automatically moved from the cloud storage tier to the system pool when the system pool reaches a certain threshold level. A file can be automatically recalled from the cloud storage tier based on some characteristics of the file such as age, size, last access time, path. Alternatively, the user can manually recall specific files or file sets. For more information on policy-based recall, see [“Applying a policy on a Transparent cloud tiering node” on page 877](#).

You can enable or disable transparent recall for a container when a container pair set is created. For more information, see [“Binding your file system or fileset to the Cloud service by creating a container pair set” on page 96](#).

Note: Like recalls in IBM Spectrum Archive and IBM Storage Protect for Space Management, Transparent cloud tiering recall would be filling the file with uncompressed data and the user would need to re-compress it by using `mmrestripefs` or `mmrestripefile` if so desired. Since we are positioning compression feature for cold data currently, the fact of a file being recalled means the file is no longer cold and leaving the file uncompressed would allow better performance for active files.

To manually trigger recall of the file `file1`, issue this command:

```
mmcloudgateway files recall file1
```

The state of the file becomes **Co-resident** after it is successfully recalled. If the file that has been recalled no longer needs to be kept in the cloud tier it can be deleted. For more information on deleting a file in the co-resident state, see [“Cleaning up files transferred to the cloud storage tier” on page 883](#).

For more information on manually recalling files, see **mmcloudgateway** command in the *IBM Storage Scale: Command and Programming Reference Guide*.

Reconciling files between IBM Storage Scale file system and cloud storage tier

This topic describes how to reconcile files that are migrated between IBM Storage Scale file systems and the cloud tier. The reconcile function runs automatically as part of maintenance activities. While it is possible to run reconcile from the CLI, it is generally not necessary to do so.

Note: To run reconcile on a given Transparent cloud tiering managed file system, ensure that enough storage capacity is available temporarily under the root file system, to allow policy scan of a file system. Rough space requirements are $(300 \times NF)$, where NF is the number of files. For example, if a Transparent cloud tiering managed file system has 1 billion inodes, then temporary space requirement for reconcile would be 300 GB $(300 \times 1\text{ B})$. For more information, see the `-s LocalWorkDirectory` option in the `mmapplypolicy` command in the *IBM Storage Scale: Command and Programming Reference Guide*.

The purpose of reconcile to is ensure that the cloud database is aligned properly with the IBM Storage Scale file system on state of files that have been tiered to the cloud. Such discrepancies can take place due to power outages and other such failures. It is recommended that this command be run every couple of months. This command needs to be run on every container pair. It should not be run in parallel with other maintenance commands like full cloud database backup but should be run in parallel with other maintenance commands (or migration policies) that affect that particular container. Also, this command should not be run while a policy migrate is being run.

There is another reason that you may want to run reconcile. Although there is a policy currently in place to automatically delete files in the cloud that have been deleted in the file system and similar support for older versions of files, that support is not fully guaranteed to remove a file. When for legal reasons or when there is a critical need to know for sure that a file has been deleted from the cloud, it is recommended that you run the reconcile command as shown below.

For example:

```
mmcloudgateway files reconcile --container-pair-set-name MyContainer gpfs.Container
Wed Nov 15 11:29:35 EST 2017
processing /ibm/gpfs.Container
Wed Nov 15 11:29:38 EST 2017 Reconcile started.
Wed Nov 15 11:29:38 EST 2017 Creating snapshot of the File System...
Wed Nov 15 11:29:39 EST 2017 Running policy on Snapshot to generate list of files to process.
Wed Nov 15 12:52:50 EST 2017 Removing snapshot.
Wed Nov 15 12:52:52 EST 2017 Reconcile is using a deletion retention period of 30 days.
Wed Nov 15 12:54:03 EST 2017 Reconcile will be processing 92617766 inode entries.
Wed Nov 15 12:54:03 EST 2017 Adding missing migrated files to the database...
Wed Nov 15 12:55:21 EST 2017 Processed 926178 entries out of 92617766.
Wed Nov 15 12:56:12 EST 2017 Processed 1852356 entries out of 92617766.
Wed Nov 15 12:56:59 EST 2017 Processed 2778533 entries out of 92617766.
Wed Nov 15 12:57:46 EST 2017 Processed 3704711 entries out of 92617766.
Wed Nov 15 12:58:34 EST 2017 Processed 4630889 entries out of 92617766.
Wed Nov 15 12:59:20 EST 2017 Processed 5557066 entries out of 92617766.

...
Wed Nov 15 14:13:15 EST 2017 Processed 92617766 entries out of 92617766.
Wed Nov 15 14:13:19 EST 2017 Reconcile found 228866 files that had been
migrated and were not in the directory.
Wed Nov 15 14:13:19 EST 2017 Reconcile detected 0 deleted files that were
deleted more than 30 days ago.
Wed Nov 15 14:13:19 EST 2017 Reconcile detected 12 migrated files that have
been deleted from the local file system, but have not been deleted from object
storage because they are waiting for their retention policy time to expire.
Wed Nov 15 14:13:19 EST 2017 Please use the 'mmcloudgateway files cloudList'
command to view the progress of the deletion of the cloud objects.
Wed Nov 15 14:13:21 EST 2017 Reconcile successfully finished.
mmcloudgateway: Command completed.
```

`gpfs.Container` is the device name of the file system that is associated with the node class, and `MyContainer` is the container where the cloud objects are stored.

You can delete files from cloud storage by using the deletion policy manager. However, you can also guarantee deletion by using a reconcile to manage the mandatory deletions. For example, if a migrated file is removed from the file system, a reconcile guarantees removal of the corresponding cloud objects and references that are contained in the cloud directory. Additionally, if multiple versions of a file are stored on the cloud, reconcile removes all older cloud versions (keeping the most recent). For example, if a file is migrated, then updated, and migrated again. In this case, two versions of the file are stored on the cloud. Reconcile removes the older version from the cloud. Reconcile also deletes cloud objects that are no longer referenced.

Note: Reconcile removes entries from the cloud directory that references deleted file system objects. Therefore, it is recommended that you restore any files that must be restored before you run a reconcile. It is also recommended to run the reconciliation operation as a background activity during low load on the Transparent cloud tiering service nodes.

Cleaning up files transferred to the cloud storage tier

This topic describes how to clean up files that are transferred to the cloud storage tier and those have not yet been deleted from the IBM Storage Scale file system.

To clean up files on cloud storage that have already been deleted from IBM Storage Scale, see “[Deleting cloud objects](#)” on page 884.

To do basic cleanup of objects that are transferred to the cloud object storage by using Transparent cloud tiering, issue a command according to this syntax:

```
mmcloudgateway files delete
{-delete-local-file | -recall-cloud-file |
--require-local-file} [--keep-last-cloud-file]
[--] File [File ...]
```

where,

- `--recall-cloud-file`: When this option is specified, the files are recalled from the cloud storage before deleting them on the cloud. The status of the local files becomes resident after the operation.

- **--delete-local-file**: This option deletes both local files and the corresponding cloud object. There is no recall here.
- **--keep-last-cloud-file**: This option deletes all the versions of the file except the last one from the cloud. For example, if a file has three versions on the cloud, then versions 1 and 2 are deleted and version 3 is retained.
- **--require-local-file**: This option removes the extended attribute from a co-resident file and makes it resident, without deleting the corresponding cloud objects. The option requires the file data to be present on the file system and will not work on a non-resident file.
- **--File**: This option can be used to process a file list similar to the one generated by the ILM policy.

The **mmcloudgateway files delete** command accepts files in GPFS file system as an input.

Note: Background maintenance automatically manages deletes from object storage for deleted or reversioned files. Hence, manual deletion is not needed.

Deleting cloud objects

The standard way to delete cloud files is to set the **--cloud-retention-period-days** setting that sets a policy that indicates how long a file must be retained after it is deleted from the file system before it must be deleted from the cloud storage tier.

Periodically, the cloud directory is scanned for deleted files by a cloud destroy utility that runs in the background. That utility checks to see which files meet the criteria of being deleted and retained longer than the cloud retention period. Files that exceed the cloud retention period days are deleted automatically by that background utility.

You can delete files by using **mmcloudgateway files delete** command or by using external commands such as **xm**. With any of these commands, the files are only deleted from the local file system, but the corresponding cloud objects are marked for deletion. These marked objects are retained on the cloud for 30 days, by default. You can modify the retention time by running the following command:

```
mmcloudgateway config set
```

After the retention period expires, the marked files are permanently deleted from the cloud storage tier.

It is recommended that you apply the destroy policy that is described because of how file deletion works. For example, when you delete files by using external commands, the cloud objects are immediately marked for deletion only if you apply the destroy policy to the file system by using the **mmchpolicy** command. If the destroy policy is not applied, the cloud objects are marked for deletion only when you run the reconcile operation. The destroy policy is available here: /opt/ibm/MCStore/samples/cloudDestroy.policy.template. Additionally, you need to apply the destroy policy along with other policies such as transparent recall and migration.

If you want to permanently delete the marked files before the retention time expires, you can use the following command:

```
mmcloudgateway files destroy
```

Run the following command to set the retention period of the cloud objects to 60 days:

```
mmcloudgateway config set --cloud-retention-period-days 60
```

You can permanently delete the cloud objects that are marked for deletion from the cloud automatically by using the destroy policy or the reconcile command.

Note: You must delete the objects only after 60 days of marking.

Run the following command if you want to delete these objects earlier than 60 days (for example, 30 days):

```
mmcloudgateway files destroy --cloud-retention-period-days 30 --container-pair-set-name
container-1
--filesystem-path /gpfs/myfold
```

Cloud objects that were marked for deletion 30 days or earlier (for files that are marked for deletion) are deleted. The cloud objects that were marked for deletion less than 30 days are retained.

For more information, see *mmcloudgateway* command in the *IBM Storage Scale: Command and Programming Reference Guide*.

Managing reverted files

Having multiple versions of a file on the cloud can be an issue especially when your storage is constrained by space and for maintenance purposes.

Cloud services manage reverted files just as how it manages the deleted files. The cloud destroy utility automatically deletes older versions depending on the retention time associated with each version.

For example, on day #1, you create a file and migrate it to the cloud. This is version 1. The retention time is NOT associated with this file yet as there are no other versions of this file. On day #2, you recall the file, modify it, and migrate it back to the cloud. This is version 2. As soon as this version is created, the retention period is applicable to the previous version (version 1). On day #6, you again recall the file, modify it, and migrate it back to the cloud. This is version 3. Once this version is created, retention period is applicable to the previous version (version 2). Now, you have a total of 3 versions of the file on your cloud storage.

The following sequence of events occurs, assuming the retention period to be 30 days:

- On day #30, a total of 3 versions of the file are there on the cloud
- On day #31, a total of 3 versions of the file are there on the cloud
- On day #32, a total of 3 versions of the file are there on the cloud
- **On day #33, version 1 is deleted as it exhausts the retention time.**
- On day #34, version 2 and 3 are there on the cloud
- On day #35, version 2 and 3 are there on the cloud
- On day #36, version 2 and 3 are there on the cloud
- **On day #37, version 2 is deleted as it exhausts the retention time.**
- On day #38, version 3 is there on the cloud

Note: There is not a separate retention policy for managing the reverted files versus deleted files. The number of days retained is the same for both as they both rely on the same policy value.

Listing files migrated to the cloud storage tier

Even if the files are deleted from the file system after migration, you can generate a list of files that are migrated to the cloud storage tier. By using the file names, you can use the **mmcloudgateway restore** option to retrieve the files back from the cloud storage tier.

To list the files that are migrated to the cloud, issue a command according to this syntax:

```
mmcloudgateway files cloudList {--path Path [--recursive [--depth Depth]] [--file File] |
--file-versions File |
--files-usage --path Path [--depth Depth] |
--reconcile-status --path Path |
--path Path --start YYYY-MM-DD[-HH:mm] --end YYYY-MM-DD[-HH:mm]}
```

Note: You can specify **--reconcile-status** only if one reconcile is running at a time. (You can run multiple reconciles in parallel, but the progress indication has this limitation.)

For example, to list all files in the current directory, issue this command:

```
mmcloudgateway files cloudList --path /gpfs0/folder1
```

To list all files in all directories under the current directory, issue this command:

```
mmcloudgateway files cloudList --path /gpfs0/folder1 --recursive
```

To find all files named *myfile* in all directories under the current directory, issue this command:

```
mmcloudgateway files cloudList --path /gpfs0/folder1 --file myfile
```

To find all files named *myfile* in the current directory, issue this command:

```
mmcloudgateway files cloudList --path /gpfs0/folder1 --depth 0 --file myfile
```

To display information about all versions of file *myfile* in current directory, issue this command:

```
mmcloudgateway files cloudList --file-versions myfile
```

Restoring files

This topic provides a brief description on how to restore files that have been migrated to the cloud storage tier if the original files are deleted from the GPFS file system.

This option provides a non-optimized (emergency) support for manually restoring files that have been migrated to the cloud storage tier if the original stub files on the GPFS file system are deleted.

Note: Transparent cloud tiering does not save off the IBM Storage Scale directory and associated metadata such as ACLs. If you want to save off your directory structure, you need use something other than Transparent cloud tiering.

Before restoring files, you must identify and list the files that need to be restored by issuing the **mmcloudgateway files cloudList** command.

Assume that the file, *afile*, is deleted from the file system but is present on the cloud, and you want to find out what versions of this file are there on the cloud. To do so, issue the following command:

```
mmcloudgateway files cloudList --file-versions /gpfs0/afile
```

The system displays output similar to this:

id	datatime	datasize	metatime	metasize	filename
13	Apr 27 03:34	6	Apr 27 03:34	499	/gpfs0/afile
14	Apr 27 03:35		12 Apr 27 03:35	693	/gpfs0/afile

You can use the output of the **cloudList** command for restoring files. For more information on the **cloudList** command, see [“Listing files migrated to the cloud storage tier” on page 885](#).

To restore files, issue a command according to this syntax:

```
mmcloudgateway files restore [-v] [--overwrite] [--restore-stubs-only]
{ -F fileListFile | [--dry-run] [--restore-location
RestoreLocation]
[ --id ID] [--] File}
```

By using this command, you can restore files in two different ways. That is, the files to be restored along with their options can be either specified at the command line, or in a separate file provided by the **-F** option.

If you want to specify the options in a file, create a file with the following information (one option per line):

```
filename=<name of the file to be retrieved>
target=<full path to restore the file>
id=<This is the unique ID that is given to each version the file (This information is available
in the cloudList output). If the id is not given, then the latest version of the file will
be retrieved.
```

The following example shows how the content needs to be provided in a file (for example, `filestoberestored`) for restoring a single file `/gpfs0/afile` with multiple versions:

```
# Restoring filename /gpfs0/afile
filename=/gpfs0/afile
target=/gpfs0/afile-33
id=33
%%
filename=/gpfs0/afile
target=/gpfs0/afile-34
id=34
%%
# Restoring filename /gpfs0/afile
filename=/gpfs0/afile
target=/gpfs0/afile-35
id=35
%%
# Restoring filename /gpfs0/afile
filename=/gpfs0/afile
target=/gpfs0/afile-latest
%%
# Restoring filename /gpfs0/afile
filename=/gpfs0/afile
```

The following example shows how the content needs to be provided in a file (for example, `filestoberestored`) for restoring the latest version of multiple files (`file1`, `file2`, and `file3`):

```
# Restoring filename /gpfs0/file1, /gpfs0/file1, and /gpfs0/file3
filename=/gpfs0/file1
target=/gpfs0/file1
%%
filename=/gpfs0/file2
target=/gpfs0/file2
%%
filename=/gpfs0/file3
target=/gpfs0/file3
```

Files to be restored are separated by lines with `%%` and `#` represents the comments.

Now that you have created a file with all required options, you need to pass this file as input to the **mmcloudgateway files restore** command, as follows:

```
mmcloudgateway files restore -F filestoberestored
```

Note: It is advised not to run the delete policy if there is some doubt that the retention policy might result in deleting of the file before you can restore it.

For information on the description of the parameters, see the **mmcloudgateway** command in *IBM Storage Scale: Command and Programming Reference Guide*.

Restoring Cloud services configuration

If your Cloud services configuration data is lost due to any unforeseen incident, you can restore the data by using the **mmcloudgateway service restoreConfig** command.

To restore the configuration data and save it to the CCR, issue a command according to the following syntax:

```
mmcloudgateway service restoreConfig --backup-config-file <name of the tar file>
```

For example, issue the following command to restore configuration data from the file, `tct_config_backup_20170915_085741.tar`:

```
mmcloudgateway service restoreConfig --backup-config-file tct_config_backup_20170915_085741.tar
```

The system displays an output similar to this:

```
You are about to restore the TCT Configuration settings to the CCR.
```

```

Any new settings since the backup was made will be lost.
The TCT servers should be stopped prior to this operation.

Do you want to continue and restore the TCT cluster configuration?
Enter "yes" to continue: yes
mmccloudgateway: Unpacking the backup config tar file...
mmccloudgateway: Completed unpacking the tar file.

Restoring the Files:
[_cloudnodeclass1.settings - Restored]
[_cloudnodeclass.settings - Restored]
[mmccloudgateway.conf - Restored]

mmccloudgateway: TCT Config files were restored to the CCR.
mmccloudgateway: Command completed.

```

Note: During the restore operation, Transparent cloud tiering servers are restarted on all the nodes.

Checking the Cloud services database integrity

There might be situations where the Cloud services database that is associated with a container gets corrupted secondary to a power outage or system crash, and in such cases, you must check the integrity of the database before you proceed with any operation.

To check the integrity of database that is associated with a container, issue a command according to the following syntax:

```
mmccloudgateway files checkDB --container-pair-set-name ContainerPairSetName
```

For example, issue the following command to check the integrity of the database that is associated with the container, container1:

```
mmccloudgateway files checkDB --container-pair-set-name container1
```

The system displays output similar to this:

```

CheckDB returned OK.
mmccloudgateway: Command completed.

```

If the database is corrupted, the system displays an output similar to this:

```

MCSTG000130E: Command Failed with following reason: Request failed with error :
Database is potentially not in a good state and needs to be rebuilt.
mmccloudgateway: Command failed. Examine previous error messages to determine cause.

```

Note: Cloud services configurations contain sensitive security-related information regarding encryption credentials, so you must store your configuration back-up in a secure location. This configuration information is critical in helping you restore your system data, if necessary.

Manual recovery of Transparent cloud tiering database

Transparent cloud tiering uses a database called "cloud directory" to store a list and versions of files that are migrated to the cloud. Any issues in this database might lead to undesired results. If the database is corrupted or has been accidentally deleted, you can reconstruct it from automatic backups of this database that are kept on the cloud.

You need to perform a recovery of the database when Transparent cloud tiering produces any of the following messages in the logs:

- The cloud directory database for file system /dev/gpfs0 could not be found. Manual recovery is necessary.
- The directory service for file system /dev/gpfs0 is not ready for use. Manual recovery is necessary.
- The cloud directory database for file system /dev/gpfs0 is corrupted. Manual recovery is necessary.

To perform a manual recovery, issue a command according to this syntax:

```
mmcloudgateway files rebuildDB --container-pair-set-name ContainerPairSetName Device
```

where,

filesystem is the device name of the file system whose database is corrupted and which is in need of manual recovery.

--container-pair-set-name is the name of the container associated with the file system or fileset.

For example, if you want to recover the database associated with the file system, /dev/gpfs0 and the container, container-1, issue this command:

```
mmcloudgateway files rebuildDB --container-pair-set container-1 /dev/gpfs0
```

The system displays output similar to this:

```
mmcloudgateway: Command completed.
```

Note: It is important that background maintenance be disabled when running this command. For more information, see the *Planning for maintenance activities* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

Scale out backup and restore (SOBAR) for Cloud services

This topic provides all necessary information for setting up and configuring SOBAR on your Cloud services cluster.

Overview

This section provides a brief introduction to SOBAR and the step-by-step instructions for backup and restore by using SOBAR.

Scale out backup and restore (SOBAR) for Cloud services is a method of disaster recovery or service migration that uses existing GPFS or SOBAR commands along with Cloud services scripts to back up configuration and file system metadata on one cluster and restore this on a recovery cluster using one sharing container pair set per node class.

Note: SOBAR works on file system boundaries, but with the Cloud services scripts, this procedure should work whether you have configured Cloud services by file system or by fileset.

The high-level steps are as follows:

Note: This procedure is designed only for data tiered to object storage by Cloud services. All other data needs to be backed up some other way.

Primary site

1. **Allocate space in object storage for the backup:** Create one sharing container pair set per Cloud services node class that is shared between the primary and recovery clusters.
 - This is used to export configuration data and metadata from the primary cluster to cloud and import to the recovery cluster.
2. **Allocate space in the associated file system for backup:** Create a global file system directory to handle the temporary space requirements of the SOBAR backup.
3. **File system configuration backup:** Back up the configuration of each file system associated with Cloud services on the primary site. If you have defined Cloud services by file set, then specify the file systems that those file sets are in. Back up the configuration of each file system on the primary site.
 - a. Securely transfer these files to a safe place
 - b. Use these to recreate compatible recovery-site file systems

4. **File system metadata backup:** Back up the Cloud services inode/metadata for file systems from a Cloud services node on the primary site using **`mcstore_sobar_backup.sh`**. If you have defined Cloud services by file set, then specify the file systems that those file sets are in.
 - This script automatically uploads the backup to the sharing container pair set on the cloud that you created earlier.
5. **Cloud services configuration backup.** Back up the Cloud services configuration data from a Cloud services node on the primary site by using the **`mmcloudgateway service backupConfig`** command.
 - Securely transfer the resulting file to a safe place.

For detailed backup instructions, see [“Procedure for backup” on page 892](#).

Recovery site

1. **Recovery site hardware and configuration preparation:** Prepare the recovery site to accommodate all the file systems that you want to recover from the primary:
 - Each file system on the recovery site must have at least as much capacity as its corresponding primary-site file system. These file systems must be created, but not mounted.

Note: You need the full space for the entire file system even if you are restoring just file set subsets - per SOBAR.

 - If you do not already have these file systems created, then you can wait until after running the **`mmrestoreconfig`** command in the subsequent step. The output generated by the **`mmrestoreconfig`** command offers guidance on how to create the recovery file system.
2. **Allocate temporary restore staging space for the file system backup image:** It is recommended to use a separate dedicated file system.
3. **File system configuration restore:** Restore the policies of each file system, fileset definitions, and other resources.
4. **Cloud services configuration restore:** Download the Cloud services configuration file (that was generated and pushed to the cloud by the **`mcstore_sobar_backup.sh`** script) using the **`mcstore_sobar_download.sh`** script on the recovery cluster.
5. **File system metadata restore:** Restore the file system metadata by running the **`mcstore_sobar_restore.sh`** script on the recovery site.

For detailed recovery instructions, see [“Procedure for restore” on page 894](#).

Note: You can recall offline files from the cloud (both manually and transparently) on the restore site only. Trying to recall offline files, migrated from the primary site, using a recall policy template does not work, because the restore site cluster does not recognize these files to be part of an external pool. However, files once migrated from the restore site can be recalled in bulk using a recall policy.

Prerequisites for using SOBAR

This topic describes the prerequisites that must be met before setting up SOBAR on your Cloud services.

Detailed preparation steps or prerequisites are as follows:

Prerequisites for the primary site

This topic describes the preparations that must be done at the primary site.

Detailed preparation steps or prerequisites are as follows:

1. Disable Cloud services maintenance operations on the appropriate node class being restored on the recovery site. For more information, see [“Configuring the maintenance windows” on page 100](#).
2. Disable all Cloud services migration policies by using the `--transparent-recalls {DISABLE}` option in the **`mmcloudgateway containerPairSet create`** command. For more information, see [“Binding your file system or fileset to the Cloud service by creating a container pair set” on page 96](#).
3. Perform the required configuration steps:

- a. Create a cloud storage account. For more information, see “[Managing a cloud storage account](#)” on page 90.
 - b. Define a cloud storage access point (CSAP). For more information, see “[Defining cloud storage access points \(CSAP\)](#)” on page 92.
 - c. Create a cloud service for Cloud data sharing by using the **mcloudgateway cloudService create** command, where you must specify --cloud-service-type as Sharing. For more information, see “[Creating Cloud services](#)” on page 94.
4. Allocate space in object storage for the backup:
- Create a shared container pair Set with a sharing cloud service and with encryption disabled (and etag enabled if a data integrity check on the SOBAR backup tar file is required). This container will be used to share the backup data between the primary site and the recovery site.
5. Calculate the maximum amount of space that will be used by the global filesystem directory (for mcstore_sobar_backup.sh script):
- a. On the primary cluster, use the **mmdf** command for each file system to determine the number of total inodes (look for Inode Information at the bottom).
- For example,
- ```
mmdf gpfs_tctbill1 | grep Inode
```
- | Inode Information                                      |           |
|--------------------------------------------------------|-----------|
| Total number of used inodes in all Inode spaces:       | 307942342 |
| Total number of free inodes in all Inode spaces:       | 748602    |
| Total number of allocated inodes in all Inode spaces:  | 308690944 |
| Total of Maximum number of inodes in all Inode spaces: | 410512384 |
- b. Add up all the used inodes for all the file systems you are backing up and multiply by 4096 to determine necessary space requirements ( $307942342 \times 4096 = 1.26\text{TB}$ ).
- Note:** This will automatically provide a buffer since the actual file will be compressed (tar) and the final size will depend on the compression ratio.
6. Allocate space in associated file system for backup: This is a global file system directory that is allocated to handle the temporary space requirements of the SOBAR backup.
- Use standard GPFS methodology or the install toolkit to allocate storage for this file system:
    - Common GPFS Principles: [“Common GPFS command principles” on page 201](#).
    - Performing additional tasks using the installation toolkit: See *Performing additional tasks using the installation toolkit* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

## **Prerequisites for the recovery site**

This topic describes the preparation steps that must be done at the secondary (restore) site.

Detailed preparation steps or prerequisites are as follows:

1. Allocate temporary restore staging space for the file system backup image:
    - This is referred to as the `global_directory_path` on the recovery site and is given the same name as the primary site in the examples.
    - It is recommended to use a separate dedicated file system.
    - Use standard GPFS methodology or the install toolkit to allocate storage for this file system:
      - Common GPFS Principles: See [“Common GPFS command principles” on page 201](#).
      - Performing additional tasks using the installation toolkit: See *Performing additional tasks using the installation toolkit* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
  2. Verify that sufficient back-end storage space exists on the recovery site for the recovered file systems:
- Note:** Each file system on the recovery site will need at least as much capacity as its corresponding primary-site file system (Actual file system creation will take place in a later step).

- On the primary cluster, use the **mmdf** command for each file system to determine the required amount of space necessary for the matching recovery site file system (look for total blocks in the second column).
- If it is necessary to determine sizes for separated metadata and data disks, look for the corresponding information on the primary site (look for data and metadata distribution in the second column). For example,

```
mmdf gpfs_tctbill1 | egrep '(data)|(metadata)|failure|fragments|total|- - - - -|====='
```

| disk<br>in KB<br>name<br>fragments | disk size<br>in KB | failure holds<br>group metadata data | holds<br>in full blocks | free in KB<br>in |
|------------------------------------|--------------------|--------------------------------------|-------------------------|------------------|
| (pool total)<br>( 0%)              | 15011648512        |                                      | 13535170560 ( 90%)      | 53986848         |
| =====                              | =====              | =====                                | =====                   | =====            |
| (data)<br>( 0%)                    | 12889330688        |                                      | 12675219456 ( 98%)      | 53886584         |
| (metadata)<br>( 0%)                | 2122317824         |                                      | 859951104 ( 41%)        | 100264           |
| =====                              | =====              | =====                                | =====                   | =====            |
| (total)<br>( 0%)                   | 15011648512        |                                      | 13535170560 ( 90%)      | 53986848         |

**Note:** NSD details are filtered out, these are displayed in 1 KB blocks (use '`--block-size auto`' to show in human readable format).

- Use the previous information as a guide for allocating NSDs on the recovery site and preparing stanza files for each file system.

**Note:** It is preferable to have the same number, size, and type of NSD for each file system on the recovery site as on the primary site, however it is not a requirement. This simply makes the auto-generated stanza file easier to modify in the recovery portion of this process.

- Ensure that there are no preexisting Cloud services node classes on the recovery site, and that the node classes that you create are clean and unused.
- Create Cloud services node classes on the recovery site by using the same node class name as the primary site. For more information, see the *Creating a user-defined node class for Transparent cloud tiering or Cloud data sharing* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
- Install (or update) Cloud services server rpm on all Cloud services nodes on the recovery site. For more information, see the *Installation steps* topic in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
- Enable Cloud services on the appropriate nodes of the recovery-site. For example,

```
mmchnode --cloud-gateway-enable -N <tctnode_ip1,tctnode_ip2,tctnode_ip3,tctnode_ip4>
--cloud-gateway-nodeclass TCTNodeClassPowerLE
```

For more information, see “[Designating the Cloud services nodes](#)” on page 88.

- Ensure that there is no active Cloud services configuration on the recovery site.
- If this is an actual disaster, and you are transferring ownership of the Cloud services to the recovery cluster, ensure that all write activity is suspended from the primary site while the recovery site has ownership of Cloud services.

## Procedure for backup

This topic describes the procedure for backing up the cluster configuration on the primary site.

Before you begin, ensure the following:

- No data migration is in progress or initiated. Starting SOBAR backup during any migration operation might eventually lead to data loss.

- The prerequisites are met and the preparation steps for the recovery site are performed. For more information, see “[Prerequisites for the primary site](#)” on page 890.

Perform the following steps:

### **1. File system configuration backup:**

- Back up the cluster configuration of every file system on the primary site that you want to recover: `/usr/lpp/mmfs/bin/mmbbackupconfig <file_system_name> -o <cluster_backup_config_file>`. For example,

```
mmbbackupconfig gpfs_tctbill1 -o powerleBillionBack_gpfs_tctbill1_02232018
```

```
mmbbackupconfig: Processing file system gpfs_tctbill1 ...
mmbbackupconfig: Command successfully completed
```

For example, to list the backup files, issue this command.

```
ls -l /root/powerleBillionBack_gpfs_tctbill*
```

```
-rw-r--r--. 1 root root 19345 Feb 23 11:24 /root/powerleBillionBack_gpfs_tctbill1_02232018
-rw-r--r--. 1 root root 19395 Feb 23 11:25 /root/powerleBillionBack_gpfs_tctbill3_02232018
```

- Securely transfer the backup files to a safe location (they will be used later in the restore process on the recovery site).

### **2. File system metadata backup (Back up the cloud data and automatically export it to the shared container).**

- From a node in your Cloud services node class on your primary site, run the `mcstore_sobar_backup.sh` script under the `/opt/ibm/MCStore/scripts` folder.

**Note:** Make sure the `<global-filesystem-directory>` you choose is mounted, accessible from all nodes in the cluster, and has enough space to accommodate the backup.

The following is an example and a sample output:

```
[root@primary-site-tct-node scripts]# /opt/ibm/MCStore/scripts/ mcstore_sobar_backup.sh
gpfs_tctbill1c powerleSOBAR1 TCTNodeClassPowerLE /ibm/gpfs_tctbill1
Creating backup for File System : gpfs_tctbill1
TOTAL_USED_INODE_SPACE 1261342920704
...
mmimgbackup: [I] Image backup of /dev/gpfs_tctbill1 begins at Wed Mar 14 17:03:05 EDT
2018.
...
mmimgbackup: [I] Image backup of /dev/gpfs_tctbill1 ends at Wed Mar 14 22:37:55 EDT 2018.
...
Exporting SOBAR backup: 9277128909880390775_gpfs_tctbill1_03-14-18-17-03-01.tar to cloud
and Data Container is : powerleSOBAR1
...
Completed backup procedure for File System : gpfs_tctbill1 use
9277128909880390775_gpfs_tctbill1_03-14-18-17-03-01.tar for restore operation
```

- Repeat the `mcstore_sobar_backup.sh` command for each file system you are backing up, using the same `sharing_container_pair_set_name` and the same `global-filesystem-directory` and the `tct_node-class-names` where appropriate.

### **3. Cloud services configuration backup**

- Issue this command: `mmcloudgateway service backupConfig --backup-file <backup_file>`. For example,

```
[root@primary-site-tct-node ~]# mmcloudgateway service backupConfig --backup-file
/temp/TCT_backupConfig
mmcloudgateway: Starting backup
Backup Config Files:
```

```

[mmcloudgateway.conf - Retrieved]
[-tctkeystore.jceks - Retrieved]
[-tctnodeclasspowerle.settings - Retrieved]
[* .p12 files - Not Found]

mmcloudgateway: Creating the backup tar file...
mmcloudgateway: Backup tar file complete.
The file is '/temp/TCT_backupConfig_20180306_123302.tar'.
mmcloudgateway: The backup file should be archived in a safe and secure location
as it may include authentication credentials.
mmcloudgateway: Command completed.

```

The backup file is located here: /temp/TCT\_backupConfig\_20180306\_123302.tar. For more information, see [“Backing up the Cloud services configuration” on page 99](#).

- Securely transfer this backup file to a safe location (It will be used later in the restore process on the recovery site).

## Procedure for restore

This topic describes the procedure for restoring data and the configuration.

Before you begin, ensure that the prerequisites are met and the preparation steps for the recovery site are performed. For more information, see [“Prerequisites for the recovery site” on page 891](#).

Perform the following steps:

- Securely transfer (by using scp or other means) the cluster configuration backup files from each file system that were generated by the **mmbackupconfig** command on the primary site to a known location on the recovery site.

To list the files on the primary site:

```

root@primary-site ~]# ls -l /root/powerleBillionBack_gpfs_tctbill*
-rw-r--r--. 1 root root 19345 Feb 23 11:24 /root/powerleBillionBack_gpfs_tctbill1_02232018
-rw-r--r--. 1 root root 19395 Feb 23 11:25 /root/powerleBillionBack_gpfs_tctbill3_02232018

```

- Transfer the `cluster_backup_config` files for each file system to the recovery cluster, as follows:

**Note:** If NSD servers are used, then transfer the backups to one of them.

```

[root@primary-site ~]# scp powerleBillionBack_gpfs_tctbill1_02232018
root@recovery-site-nsd-server-node:/temp/ powerleBillionBack_gpfs_tctbill1_02232018

```

```

scp powerleBillionBack_gpfs_tctbill3_02232018 root@recovery-site-nsd-server-node:/temp/
powerleBillionBack_gpfs_tctbill1DB_02232018

```

### 2. File system configuration restore

- Create the file system configuration restore file `restore_out_file` for each file system on the recovery site, as follows:

```

mmrestoreconfig <file-system> -i <cluster_backup_config_file> -F <restore_out_file>

```

For example,

```

[root@ recovery-site-nsd-server-node ~]# mmrestoreconfig gpfs_tctbill1 -i
/roggr/powerleBillionBack_gpfs_tctbill1_02232018 -F
./powerleBillionRestore_gpfs_tctbill1_02232018

```

```

mmrestoreconfig: Configuration file successfully created in
./powerleBillionRestore_gpfs_tctbill1_02232018
mmrestoreconfig: Command successfully completed

```

The <restore\_out\_file> (powerleBillionRestore\_gpfs\_tctbill1\_02232018 in this example) that is populated by the **mmrestoreconfig** command creates detailed stanzas for NSDs as they are on the primary site (these will need to be modified to match the NSD configuration on the recovery site). It also contains a detailed **mmcrls** command that can be used to create the associated file system on the recovery site.

**Note:** Disable Quota (remove the -Q yes option from this command) when you run it later in the process.

Some excerpts from the `restore_out_file` (powerleBillionBack\_gpfs\_tctbill1\_02232018):

```

Filesystem configuration file backup for file system: gpfs_tctbill1
Date Created: Tue Mar 6 14:15:05 CST 2018
##
The '#' character is the comment character. Any parameter
modified herein should have any preceding '#' removed.

NSD configuration
Disk descriptor format for the mmcrls command.
Please edit the disk and desired name fields to match
your current hardware settings.
##
The user then can uncomment the descriptor lines and
use this file as input to the -F option.
##
...
Then it lists all the nsds (15 of them in this case):
%nsd:
device=DiskName
nsd=nsd11
usage=dataOnly
failureGroup=1
pool=system
#
%nsd:
device=DiskName
nsd=nsd12
usage=dataOnly
failureGroup=1
pool=system
#
etc.....

%pool:
pool=system
blockSize=4194304
usage=dataAndMetadata
layoutMap=scatter
allowWriteAffinity=no
#
File system configuration
The user can use the predefined options/option values
when recreating the filesystem. The option values
represent values from the backed up filesystem.
##
mmcrls FS_NAME NSD_DISKS -i 4096 -j scatter -k nfs4 -n 100 -B 4194304 -Q yes
-version 5.0.0.0 -L 33554432 -S relatime -T /ibm/gpfs_tctbill1 --inode-limit
407366656:307619840
#
When preparing the file system for image restore, quota
enforcement must be disabled at file system creation time.
If this is not done, the image restore process will fail.
...
Disk Information
Number of disks 15
nsd11 991486976
nsd12 991486976
etc.....
nsd76 1073741824

Number of Storage Pools 1
system 15011648512
etc...
Policy Information

```

```
/* DEFAULT */
/* Store data in the first data pool or system pool */

Fileset Information
NFS_tctbill1 Linked /ibm/gpfs_tctbill1/NFS_tctbill1 off Comment:
etc...

Quota Information
Type Name USR root
etc...
```

Example portion of modified nsd stanzas for <restore\_out\_file>:

```
%nsd:
device=/dev/mapper/mpaths
nsd=nsd47
servers=nsdServer1,nsdServer2
usage=dataOnly
failureGroup=1
pool=system

%nsd:
device=/dev/mapper/mpath1
nsd=nsd48
servers= nsdServer2,nsdServer1
usage=dataOnly
failureGroup=1
pool=system

%nsd:
device=/dev/mapper/mpathbz
nsd=nsd49
servers= nsdServer1,nsdServer2
usage=metadataOnly
failureGroup=1
pool=system
```

- b. Modify the *restore\_out\_file* to match the configuration on the recovery site. Example portion of modified nsd stanzas for *restore\_out\_file* is as follows:

```
%nsd:
device=/dev/mapper/mpaths
nsd=nsd47
servers=nsdServer1,nsdServer2
usage=dataOnly
failureGroup=1
pool=system

%nsd:
device=/dev/mapper/mpath1
nsd=nsd48
servers= nsdServer2,nsdServer1
usage=dataOnly
failureGroup=1
pool=system

%nsd:
device=/dev/mapper/mpathbz
nsd=nsd49
servers= nsdServer1,nsdServer2
usage=metadataOnly
failureGroup=1
pool=system
```

### 3. Create recovery-site NSDs if necessary.

- a. Use the newly modified *restore\_out\_file* (powerleBillionRestore\_gpfs\_tctbill1\_02232018\_nsd in this example) to create NSDs on the recovery cluster. This command must be run from an NSD server node (if NSD servers are in use):

```
[root@recovery-site-nsd-server-node ~]# mmcrlnsd -F
/temp/powerleBillionRestore_gpfs_tctbill1_02232018_nsd
mmcrlnsd: Processing disk mapper/mpathq
```

```

etc...
mmcrnsd: Processing disk mapper/mpathcb
mmcrnsd: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.

```

- b. Repeat the **mmcrnsd** command appropriately for each file system that you want to recover.

#### 4. Create recovery-site file systems if necessary.

- a. Use the same modified *restore\_out\_file*

(powerleBillionRestore\_gpfs\_tctbill1\_02232018\_nsd in this example) as input for the **mmcrfs** command, which will create the file system. The following example is based on the command included in the <restore\_out\_file> (note the '-Q yes' option has been removed). For example,

```

root@recovery-site-nsd-server-node ~]# mmcrfs gpfs_tctbill1 -F
/temp/powerleBillionRestore_gpfs_tctbill1_02232018_nsd
-i 4096 -j scatter -k nfs4 -n 100 -B 4194304 --version 5.0.0.0 -L 33554432 -S
relatime -T /ibm/gpfs_tctbill1 --inode-limit 407366656:307619840

```

- b. Repeat the **mmcrfs** command appropriately for each file system that you want to recover.

#### 5. Cloud services configuration restore (download SOBAR backup from the cloud for the file system).

- a. Securely transfer the Cloud services configuration file to the desired location by using **scp** or any other commands.

- b. From the appropriate Cloud services server node on the recovery site (a node from the recovery Cloud services node class), download the SOBAR.tar by using the **mcstore\_sobar\_download.sh** script. This script is there in the /opt/ibm/MCStore/scripts folder on your Cloud services node.

**Note:** Make sure your *local\_backup\_dir* is mounted and has sufficient space to accommodate the SOBAR backup file. It is recommended to use a GPFS file system.

```

Usage: mcstore_sobar_download.sh <tct_config_backup_path>
<sharing_container_pairset_name>
<node-class-name> <sobar_backup_tar_name> <local_backup_dir>

```

For example,

```

[root@recovery-site-tct-node scripts]# ./mcstore_sobar_download.sh
/temp/TCT_backupConfig_20180306_123302.tar powerleSOBAR1 TCTNodeClassPowerLE
9277128909880390775_gpfs_tctbill1_03-14-18-17-03-01.tar /ibm/gpfs_tct_SOBAR1/

```

```

You are about to restore the TCT Configuration settings to the CCR.
Any new settings since the backup was made will be lost.
The TCT servers should be stopped prior to this operation.

```

```

Do you want to continue and restore the TCT cluster configuration?
Enter "yes" to continue: yes

```

```

mmcloudgateway: Unpacking the backup config tar file...
mmcloudgateway: Completed unpacking the tar file.

```

```

Restoring the Files:
[mmcloudgateway.conf - Restored]
[_tctkeystore.jceks - Restored]
[_tctnodeclasspowerle.settings - Restored to version 96]

```

```

mmcloudgateway: TCT Config files were restored to the CCR.
mmcloudgateway: Command completed.
mmcloudgateway: Sending the command to node recovery-site-tct-node.
Stopping the Transparent Cloud Tiering service.
mmcloudgateway: The command completed on node recovery-site-tct-node.

```

```

mmcloudgateway: Sending the command to node recovery-site-tct-node2.
Stopping the Transparent Cloud Tiering service.
mmcloudgateway: The command completed on node recovery-site-tct-node2.
mmcloudgateway: Command completed.

```

```

etc...
mmcloudgateway: Sending the command to node recovery-site-tct-node.
Starting the Transparent Cloud Tiering service...
mmcloudgateway: The command completed on node recovery-site-tct-node.

etc...

Making sure Transparent Service to start on all nodes.
Please wait as this will take some time..

Downloading 9277128909880390775_gpfs_tctbill1_03-14-18-17-03-01.tar from cloud.
This will take some time based on the size of the backup file.
Please wait until download completes..
Download of 9277128909880390775_gpfs_tctbill1_03-14-18-17-03-01.tar from cloud
completed successfully.
Moving Backup tar 9277128909880390775_gpfs_tctbill1_03-14-18-17-03-01.tar under
/ibm/gpfs_tct_SOBAR1/
Note: Before running mcstore_sobar_restore.sh to restore the file system metadata,
make sure that file system to be restored is clean and never been mounted for write.

```

## 6. File system configuration restore (Restore file system configuration on the recovery site)

**Note:** If your temporary restore staging space is on a Cloud services managed file system, then you will have to delete and recreate this Cloud services managed file system at this point.

- Restore policies for each file system using the **mmrestoreconfig** command.

```
Usage: mmrestoreconfig Device -i InputFile --image-restore
```

For example,

```
[root@recovery-site-tct-node]# mmrestoreconfig gpfs_tctbill1 -i
/temp/powerleBillionBack_gpfs_tctbill1_02232018 --image-restore

Configuration restore of gpfs_tctbill1 begins at Fri Mar 16 05:48:06 CDT 2018.

mmrestoreconfig: Checking disk settings for gpfs_tctbill1:
mmrestoreconfig: Checking the number of storage pools defined for gpfs_tctbill1.
mmrestoreconfig: Checking storage pool names defined for gpfs_tctbill1.
mmrestoreconfig: Checking storage pool size for 'system'.

mmrestoreconfig: Checking filesystem attribute configuration for gpfs_tctbill1:
mmrestoreconfig: Checking fileset configurations for gpfs_tctbill1:
Fileset NFS_tctbill1 created with id 1 root inode 536870915.
Fileset NFS_tctbill1_bkg created with id 2 root inode 1073741827.
Fileset NFS_tctbill1_bkg1 created with id 3 root inode 1610612739.

mmrestoreconfig: Checking policy rule configuration for gpfs_tctbill1:
Restoring backed up policy file.
Validated policy 'policyfile.backup':
Policy 'policyfile.backup' installed and broadcast to all nodes.
mmrestoreconfig: Command successfully completed
```

## 7. Restore file system metadata using the **mcstore\_sobar\_restore.sh** script found in the **/opt/ibm/MCStore/scripts** folder. The **mcstore\_sobar\_restore.sh** script does the following:

- The **mcstore\_sobar\_restore.sh** script does the following:

- Untars the **sobar\_backup\_file**
- Stops the Cloud services for the specified node class
- Unmounts the recovery file system and re-mounts read-only
- Restores the recovery file system image
- Re-mounts the recovery file system in read/write
- Enables and restarts Cloud services
- Executes the file curation policy - changing objects from Co-Resident state to Non-Resident state
- Rebuilds the Cloud services database files if you choose to do so

**Note:** If Cloud directory is pointing to another file system, make sure that the file system is mounted correctly before you run the restore script providing the rebuildDB parameter value to yes.

```
[root@recovery-site-tct-node scripts]# ./mcstore_sobar_restore.sh
/ibm/gpfs_tct_SOBAR1/9277128909880390775_gpfs_tctbill1_03-14-18-17-03-01.tar gpfs_tctbill1
TCTNodeClassPowerLE yes /ibm/gpfs_tct_SOBAR1 >> /root/status.txt

etc...

[I] RESTORE:[I] This task restored 1310720 inodes
[I] A total of 307 PDRs from filelist /dev/null have been processed; 0 'skipped' records
and/or errors.
[I] Finishing restore with conclude operations.
[I] CONCLUDE:[I] Starting image restore pipeline
[I] A total of 307 files have been migrated, deleted or processed by an
EXTERNAL EXEC/script;
0 'skipped' files and/or errors.
Fri Mar 16 17:20:29 CDT 2018: mmumount: Unmounting file systems ...
Fri Mar 16 17:20:33 CDT 2018: mmmount: Mounting file systems ...

etc.....

Running file curation policy and converting co-resident files to Non resident.
This will take some time. Please wait until this completes..

[I] GPFS Current Data Pool Utilization in KB and %
Pool_Name KB_Occupied KB_Total Percent_Occupied
system 327680 12287688704 0.002666734%
[I] 307944153 of 410512384 inodes used: 75.014583%.
[I] Loaded policy rules from /opt/ibm/MCStore/samples/CoresToNonres.sobar.template.
Evaluating policy rules with CURRENT_TIMESTAMP = 2018-03-16@22:24:28 UTC
Parsed 2 policy rules.

etc...
Completed file curation policy execution of converting co-resident files to
Non resident files.
running rebuild db for all the tiering containers for the given file system :
gpfs_tctbill1
Running rebuild db for container pairset : powerlebill1spill2 and File System:
gpfs_tctbill1
mmcloudgateway: Command completed.
Running rebuild db for container pairset : powerlebill1spill1 and File System:
gpfs_tctbill1
mmcloudgateway: Command completed.
Running rebuild db for container pairset : powerlebill1 and File System: gpfs_tctbill1
etc...
```

- b. Repeat the **mcstore\_sobar\_restore.sh** script appropriately for each file system that you want to recover.
- 8. Enable Cloud services maintenance operations on the appropriate node class being restored on the recovery site. For more information, see [“Configuring the maintenance windows” on page 100](#).
- 9. Enable all Cloud services migration policies on the recovery site by using the --transparent-recalls {ENABLE} option in the **mmcloudgateway containerPairSet update** command. For more information, see [“Binding your file system or fileset to the Cloud service by creating a container pair set” on page 96](#).

## Description of file names and parameters used in the example

### Primary site

Command: **mmbbackupconfig**

Usage: **mmbbackupconfig Device -o OutputFile**

| Table 69. Parameter description |                                         |
|---------------------------------|-----------------------------------------|
| Name                            | Description                             |
| file_system_name (Device)       | Name of the file system to be backed up |

Table 69. Parameter description (continued)

| Name                                       | Description                                                                                                                                                                                                                                                                                                       |
|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| filesystem_backup_config_file (OutputFile) | A unique file name that holds information for a specific backed-up file system: <ul style="list-style-type: none"> <li>• You will create a unique name (filesystem_backup_config_file) for each backed-up file system (which means you run this command once for each file system you want to back up)</li> </ul> |

Command: **mcstore\_sobar\_backup.sh**

Usage: **mcstore\_sobar\_backup.sh <file\_system\_names> <sharing\_container\_pairset\_name> <node\_class\_name> <global\_filesystem\_directory>**.

Table 70. Parameter description

| Name                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| file_system_names               | A comma-separated list of file systems to back up when using the mcstore_backup.sh script: <ul style="list-style-type: none"> <li>• As of IBM Storage Scale release 5.0.1.0, using multiple file names in this script will run backups in series.</li> <li>• Cloud services nodes chosen automatically share the workload for creating the backups.</li> </ul>                                                                  |
| sharing_container_pair_set_name | The name of a shared cloud container that is created and is accessed by both the primary and recovery clusters. This container must be large enough to accommodate space that is calculated according to this formula: 4 KB x number of inodes of all backed-up file systems. For example, if the number of inodes of the backed-up file system is 1000, then the recommended size of the container should be (4x1000)=4000 KB. |
| node_class_names                | The names of the Cloud services node classes associated with this backup.                                                                                                                                                                                                                                                                                                                                                       |
| global_filesystem_directory     | This is a working directory that is used to store data for the backups: <ul style="list-style-type: none"> <li>• It is recommended to use a directory of a GPFS file system that is accessible by all nodes in the cluster to avoid possible local root file system overload.</li> <li>• It is also acceptable to use the GPFS file system that is being backed up (as long as sufficient space exists).</li> </ul>             |

Command: **mmcloudgateway service backupConfig --backup-file <BackupFile>**, where *BackupFile* is a file used specifically for backing up all the Cloud services configuration data of all the node classes on the primary site.

### Recovery Site

**Command:** `mmrestoreconfig`

*Usage (to create a restore\_out\_file):* `mmrestoreconfig Device -i InputFile -F QueryResultFile`

*Usage (to restore image):* `mmrestoreconfig Device -i InputFile --image-restore`

*Table 71. Parameter description*

| Name                          | Description                                                                                    |
|-------------------------------|------------------------------------------------------------------------------------------------|
| file_system                   | Name of file systems to be recovered (matching backed up file systems)                         |
| filesystem_backup_config_file | The file systems backups that you created on the primary and transferred to the recovery site. |

**Command:** `mcstore_sobar_download.sh`

*Usage:* `mcstore_sobar_download.sh <tct_config_backup_path> <sharing_container_pairset_name> <node_class_name> <sobar_backup_tar_name> <local_backup_dir>`

*Table 72. Parameter description*

| Name                           | Description                                                                                                                                                                                                                        |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tct_backup_config_path         | Path on the recovery site that has the Cloud services backup tar file that was generated with the <b>mmccloudgateway service backupConfig</b> command and securely transferred to a recovery site Cloud services node by the user. |
| sharing_container_pairset_name | The sharing container that you created as a prerequisite to this procedure.                                                                                                                                                        |
| node-class-name                | The name of the Cloud services node class that is restored.                                                                                                                                                                        |
| sobar_backup_tar_name          | The name of the .tar file that was generated by the <b>mcstore_sobar_backup.sh</b> script on the primary site, and transferred to the sharing container.                                                                           |
| local_backup_dir               | A directory of your choice that is large enough to accept the 'SOBAR'.tar file. It is recommended to use a GPFS file system.                                                                                                       |

**Command:** `mcstore_sobar_restore.sh`

*Usage:* `mcstore_sobar_restore.sh <sobar_backup_path> <file_system_name> <node_class_name> <rebuilddb_required: yes/no> <global_filesystem_directory>`

*Table 73. Parameter description*

| Name              | Description                                                                           |
|-------------------|---------------------------------------------------------------------------------------|
| sobar_backup_path | Path to the 'SOBAR'.tar as designated by the <b>mcstore_sobar_download.sh</b> script. |
| file_system_name  | File system that is being restored                                                    |
| node_class_name   | Name of the Cloud services node class that is being restored.                         |

Table 73. Parameter description (continued)

| Name                        | Description                                                                                                                                                                                                           |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| rebuilddb_required          | Yes/no if a rebuild of the Cloud services metadata database is required. If you have Cloud services metadata database file systems separated from your data-only file systems, you will need to back them up as well. |
| global_filesystem_directory | The path to the file system that is shared with the primary site via the sharing container pair set.                                                                                                                  |

## Cloud data sharing

You can share data between storage servers by using the import and export function available in IBM Storage Scale.

Cloud data sharing works by combining the import and export functions that allow data to be moved across disparate geographical locations and/or heterogeneous application platforms. Cloud data sharing maintains a set of records of those moves called a manifest that enable applications to know what has moved. An application at one site can generate data, export it to the cloud, and applications at other sites can import and process that data. Applications can know what data has moved and is, therefore, now available by looking at the manifest file. It is also a way to easily move data back and forth between local and cloud storage systems. Cloud data sharing supports moving data to the cloud and pulling data from the cloud. Cloud data sharing must be configured with a local file system and a cloud account. Once configured, data can be moved between the IBM Storage Scale file system and the cloud account.

### Application considerations

Exporting applications need some mechanism to both notify other applications that new data is available on the cloud and give those applications some way of understanding what objects were put to the cloud. Cloud data sharing services provide a manifest to help applications communicate that new data is available and what that data is. When data is exported, an option to build a manifest file can be specified. This manifest is a text file that contains the name of the cloud objects exported and some other information that can be used by an application that wants to import the full data, or a subsection of it.

When data is imported, there are cases in which not all the data is needed and this unneeded data can be identified by information in the file metadata. In these cases, it is recommended that as a first pass the file headers are imported only with the **import-only-stub** option. The policy engine can then be used to import only those files that are needed, thereby saving transfer time and cost. For now this import of stub includes metadata only for data that was previously exported by IBM Storage Scale.

**Note:** For many cloud services, enabling indexed containers can impact performance, so it is possible that cloud containers are not indexed. For these situations, a manifest is mandatory. But even with indexing enabled, for large containers that contain many objects, a manifest can be useful.

Additionally, this manifest utility can be used by a non-IBM Storage Scale application to build a manifest file for other applications, including IBM Storage Scale, to use for importing purposes.

There is a manifest utility that can run separate from IBM Storage Scale (it is a Python script) that can be used to look at the manifest. It provides a way to list and filter the manifest content, providing comma separated value output.

### An overview of using import and export CLI commands

To export files to a cloud storage tier, issue a command according to the following syntax:

```
mmcloudgateway files export
 [--tag Tag]
 [--target-name TargetName]
```

```
[--container Container | no-container]
[--manifest-file ManifestFile]
[--export-metadata [-fail-if-metadata-too-big]]
[--strip-filesystem-root]
File[File] }
```

The following example exports a local file named /dir1/dir2/file1 to the cloud and store it in a container named MyContainer. A manifest file will be created, and the object exported to the cloud will have an entry in that manifest file tagged with MRI\_Images.

```
mmcloudgateway files export --container MyContainer --tag MRI_Images --export-metadata --
manifest-file
/dir/ManifestFile /dir1/dir2/file1
```

To import files from a cloud storage tier, issue a command according to the following syntax:

```
mmcloudgateway files import
[--container Container | no-container]
[--import-only-stub]
[--import-metadata]
{ [--directory Directory] | [--directory-root DirectoryRoot] | [--target-name TargetName] }
{ PolicyFile -e | [--] File[File] }
```

The following example imports files from the cloud storage tier and creates a necessary local directory structure.

```
mmcloudgateway files import --directory /localdir /dir1/dir2/file1
```

For more information on the usage of the import and export functions, see the [mmcloudgateway](#) man page.

## Listing files exported to the cloud

This topic describes how to parse a manifest file and how to list files from the cloud.

Although files are exported to the cloud from the IBM Storage Scale environment, the files can be imported by a non-IBM Storage Scale application. While you export files to the cloud, a manifest file is built. The manifest file includes a list of these exported files and the metadata associated with native object storage.

When data is exported to the cloud, the manifest file is not automatically pushed to the cloud. You must decide when and where to export the manifest file.

**When to transfer:** If you are using a policy to export data, a good time to export the manifest is immediately after the policy has successfully executed your executive chain. Waiting too long can result in manifest that is too big and that does not provide frequent enough guidance to applications looking for notifications about new data on the cloud. Constantly pushing out new manifests can create other problems where the applications have to deal with many small manifests, and having to understand which they should use.

**Where to transfer:** Unlike transparent cloud tiering, cloud data sharing allows data to be transferred to any container at any time. This freedom can be very useful, especially when setting up multiple tenants. A centralized manifest is useful in a single tenant environment, but when there are multiple tenants with different access privileges to different files it may be better to split up your manifest destinations accordingly. Export all data targeted to a particular tenant and then send the manifest. Export data for the next tenant, and so forth.

The manifest file is a text file whose entry format is as follows:

```
<File/Object Name> <CloudContainerName> <TagID> <TimeStamp><Newline>
```

Typically, this file is not accessed directly but rather is accessed using the manifest utility.

A manifest utility produces a CSV stream entry format is as follows:

```
<TagID>,<CloudContainerName>,<TimeStamp>,<File/Object Name><newline>
```

where,

- TagID is an optional identifier the object is associated with.
- CloudContainerName is the name of the container the object was exported into.
- TimeStamp follows the format: "DD MON YYYY HH:MM:SS GMT".
- File/Object Name can contain commas, but not new line characters.

An example entry in a manifest utility stream output is as follows:

```
0, imagecontainer, 6 Sep 2016 20:31:45 GMT, images/a/cat.scan
```

You can use the **mmcloudmanifest** tool to parse the manifest file that is created by the **mmcloudgateway files export** command or by any other means. By looking at the manifest files, an application can download the desired files from the cloud.

The **mmcloudmanifest** tool is automatically installed on your cluster along with Transparent cloud tiering rpms. However, you must install the following packages for the tool to work:

- Install Python version 3.6
- Install pip. For more information, see [https://packaging.python.org/install\\_requirements\\_linux/](https://packaging.python.org/install_requirements_linux/)
- Install apache-libcloud package by running the **sudo pip install apache-libcloud** command.

**Note:** Only while working with Swift3, installing latest version of **apache-libcloud** might not work. Hence, run **pip install apache-libcloud==1.3.0** to install the specific version to address the dependency.

The syntax of the tool is as follows:

```
mmcloudmanifest
ManifestName [--cloud --properties-file PropertiesFile --manifest-container ManifestContainer
[--persist-path PersistPath]
[--tag-filter TagFilter] [--container-filter ContainerFilter]
[--from-time FromTime] [--path-filter PathFilter]
[--help]
```

where,

- **ManifestName**: Specifies the name of the manifest object that is there on the cloud. For using a local manifest file, specify the full path name to the manifest file.
- **--properties-file PropertiesFile**: Specifies the location of the properties file to be used when retrieving the manifest file from the cloud. A template properties file is located at /opt/ibm/MCStore/scripts/provider.properties. This file includes details such as the name of the cloud storage provider, credentials, and URL.
- **--persist-path PersistPath**: Stores a local copy of the manifest file that is retrieved from the cloud in the specified location.
- **--manifest-container ManifestContainer**: Name of the container in which the manifest is located.
- **--tag-filter TagFilter**: Lists only the entries whose Tag ID # matches the specified regular expression (regex).
- **--container-filter ContainerFilter**: Lists only the entries whose container name matches the specified regex.
- **--from-time FromTime**: Lists only the entries that occur starting at or after the specified time stamp. The time stamp must be enclosed within quotations, and it must be in the 'DD MON YYYY HH:MM:SS GMT' format. Example: '21 Aug 2016 06:23:59 GMT'

- **--path-filter PathFilter:** Lists only the entries whose path name matches the specified regex.

The following command exports four CSV files tagged with "us-weather", along with the manifest file, "manifest.txt", to the cloud:

```
mmcloudgateway files export --container arn8781724981111500553 --manifest-file manifest.txt
--tag us-weather /gpfs/weather_data/MetData_Oct06-2016-Oct07-2016-ALL.csv
/gpfs/weather_data/MetData_Oct07-2016-Oct08-2016-ALL.csv
/gpfs/weather_data/MetData_Oct08-2016-Oct09-2016-ALL.csv
/gpfs/weather_data/MetData_Oct09-2016-Oct10-2016-ALL.csv
/gpfs/weather_data/MetData_Oct10-2016-Oct11-2016-ALL.csv
```

The following command exports four CSV files tagged with "uk-weather", along with the manifest file, "manifest.txt", to the cloud:

```
mmcloudgateway files export --container arn8781724981111500553 --manifest-file manifest.txt
--tag uk-weather /gpfs/weather_data/MetData_Oct06-2016-Oct07-2016-ALL.csv
/gpfs/weather_data/MetData_Oct07-2016-Oct08-2016-ALL.csv
/gpfs/weather_data/MetData_Oct08-2016-Oct09-2016-ALL.csv
/gpfs/weather_data/MetData_Oct09-2016-Oct10-2016-ALL.csv
/gpfs/weather_data/MetData_Oct10-2016-Oct11-2016-ALL.csv
```

So, the container "arn8781724981111500553" contains both US and UK weather data.

The following command parses the manifest file and imports the files that are tagged with "us-weather" to the local file system under the /gpfs directory:

```
mmcloudmanifest parse-manifest manifest.txt --tag-filter us-weather
| xargs mmcloudgateway files import --directory /gpfs --container arn8781724981111500553
```

You can verify these files by using the following command:

```
ls -l /gpfs
```

The system displays output similar to this:

```
total 64
drwxr-xr-x. 2 root root 4096 Oct 5 07:09 automountdir
-rw-r--r--. 1 root root 7859 Oct 18 02:15 MetData_Oct06-2016-Oct07-2016-ALL.csv
-rw-r--r--. 1 root root 7859 Oct 18 02:15 MetData_Oct07-2016-Oct08-2016-ALL.csv
-rw-r--r--. 1 root root 14461 Oct 18 02:15 MetData_Oct08-2016-Oct09-2016-ALL.csv
-rw-r--r--. 1 root root 14382 Oct 18 02:15 MetData_Oct09-2016-Oct10-2016-ALL.csv
-rw-r--r--. 1 root root 14504 Oct 18 02:15 MetData_Oct10-2016-Oct11-2016-ALL.csv
drwxr-xr-x. 2 root root 4096 Oct 17 14:12 weather_data
```

## Importing cloud objects exported through an old version of Cloud data sharing

Use the following procedure for situations where you export files to the cloud storage tier by using Cloud services 4.2.3.x (before upgrade) and then want to import those files by using Cloud services 5.2.1 (after upgrade).

1. If the container includes migrated files, clean up the associated fileset or file system objects from the container.
2. Run the following command to delete the container pair set that is associated with the container in the new version of Transparent cloud tiering:

```
mmcloudgateway containerPairSet delete
```

3. Run the following command to create a Cloud data sharing service by using the following command:

```
mmcloudgateway cloudService create
```

4. Use the sharing service that is created in the previous step and run the following command. Running this command creates a container pair set that points to the same container name as the container in the previous version of Transparent cloud tiering:

```
mmcloudgateway containerPairSet create
```

**Note:** You need to include the following parameters while you create a container pair set when encryption is enabled in the older release:

- KeyManagerName
- ActiveKey

5. Run the following command to import the files:

```
mmcloudgateway files import
```

## Administering Transparent cloud tiering and Cloud data sharing services

This topic provides a brief description on how to manage Transparent cloud tiering and Cloud data sharing in the IBM Storage Scale cluster.

### Stopping Cloud services software

This topic describes the procedure for stopping the Cloud services software.

To stop Cloud services on all Transparent cloud tiering nodes in a cluster, issue the following command:

```
mmcloudgateway service stop -N alltct
```

To stop the Cloud services on a specific node or a list of nodes, issue a command according to this syntax:

```
mmcloudgateway service stop [-N alltct {Node[,Node...] | NodeFile | NodeClass}]
```

For example, to stop the service on the node, 10.11.12.13, issue this command:

```
mmcloudgateway service stop -N 10.11.12.13
```

You can run this command on any node in the cluster.

**Note:** Before you stop Cloud services, ensure that no migration or recall operation is running on the system where the service is stopped. You can find out the status of the migration or recall operation from the GUI metrics.

### Monitoring the health of Cloud services software

Use the mmcloudgateway command to monitor the health of cloud services.

To monitor the status of Cloud services, issue a command according to this syntax:

```
mmcloudgateway service status [-N {alltct | Node[,Node...] | NodeFile | NodeClass}]
[--cloud-storage-access-point-name CloudStorageAccessPointName] [-Y]
```

For example,

- To check the status of all available Transparent cloud tiering nodes in your cluster, issue this command:

```
mmcloudgateway service status -N alltct
```

The system displays output similar to this:

```
Cloud Node Class: tct
=====
```

| Cloud Service | Status                 | Reason        |                              |                             |        |         |
|---------------|------------------------|---------------|------------------------------|-----------------------------|--------|---------|
| swift-service | ENABLED                |               |                              |                             |        |         |
| Node          | Daemon node name       | Server Status | Account / CSAP               | Container / File System/Set | Status | Reasons |
| 1             | vm597.pk.slabs.ibm.com | STARTED       | swift-account<br>swift-point | swift-pair<br>/gpfs/        | ONLINE |         |
| 2             | vm482.pk.slabs.ibm.com | STARTED       | swift-account<br>swift-point | swift-pair<br>/gpfs/        | ONLINE |         |

- To check the status of all available Transparent cloud tiering nodes in a specific node class (TctNode), issue this command:

```
mmcloudgateway service status -N TctNode
```

The system displays output similar to this:

| Cloud Node Class: TctNode |         |        |      |                  |               |                                             |
|---------------------------|---------|--------|------|------------------|---------------|---------------------------------------------|
| Cloud Service             | Status  | Reason | Node | Daemon node name | Server Status | Account / Container / File System/Set       |
| cs1                       | ENABLED |        | 1    | pk.slabs.ibm.com | STARTED       | swift-next<br>csapnext<br>swift-new<br>csap |
| cs2                       | ENABLED |        |      |                  |               | cpairnext<br>/gpfs/fs1/                     |
|                           |         |        |      |                  |               | ONLINE                                      |
|                           |         |        |      |                  |               | ONLINE                                      |

- To check the status of all available Transparent cloud tiering nodes in a specific CSAP, issue this command:

```
mmcloudgateway service status -N TctNode --cloud-storage-access-point-name swift-point
```

The system displays output similar to this:

| Cloud Node Class: TctNode    |                  |         |               |                 |             |         |
|------------------------------|------------------|---------|---------------|-----------------|-------------|---------|
| Cloud Service Status Reason  |                  |         |               |                 |             |         |
| swift-service ENABLED        |                  |         |               |                 |             |         |
| Server Account / Container / |                  |         |               |                 |             |         |
| Node                         | Daemon node name | Status  | CSAP          | File System/Set | Status      | Reasons |
| 1                            | jupiter-vm1192   | STARTED | swift-account | swift-pair      | swift-point | /gpfs/  |
|                              |                  |         |               |                 | Online      | ONLINE  |

**Note:** ONLINE status here means container exists on the cloud, but it does not guarantee that the migrations would work. This is because there could be storage errors on object storage, due to which new object creation might fail. To verify container status for migrations, issue the **mmcloudgateway containerpairset test** command.

For more information on all the available statuses and their description, see the *Transparent Cloud Tiering status description* topic in *IBM Storage Scale: Command and Programming Reference Guide*.

## GUI navigation

To work with this function in the GUI,

- Log on to the IBM Storage Scale GUI and select **Files >Transparent cloud tiering**
- Log on to the IBM Storage Scale GUI and select **Monitoring>Statistics**

Additionally, you can check the Cloud services status by using the **mmhealth node show CLOUDGATEWAY** command.

## Checking the Cloud services version

This topic describes how to check the Cloud services versions of each node in a node class.

CLI commands do not work on a cluster if all nodes in a node class are not running the same version of the Cloud services. For example, you have three nodes (node1, node2, node3) in a node class (TCTNodeClass1). Assume that the Cloud services version of node1 is 1.1.1, of node2 is 1.1.1, and of node3 is 1.1.2. In this case, the CLI commands specific to 1.1.2 do not work in the TCTNodeClass1 node class.

To check the service version of all Transparent cloud tiering nodes in a cluster, issue the following command:

```
mmcloudgateway service version -N alltct
```

To check for service versions associated with Cloud services nodes, issue a command according to this syntax:

```
mmcloudgateway service version [-N {Node[,Node...] | NodeFile | NodeClass}]
```

For example, to display the Cloud services version of the nodes, node1 and node2, issue the following command:

```
mmcloudgateway service version -N node1,node2
```

The system displays output similar to this:

| Node | Cloud node name | TCT Type | TCT Version |
|------|-----------------|----------|-------------|
| 8    | node1           | Client   | 1.1.5       |
| 9    | node2           | Server   | 1.1.5       |

To display the Cloud services version of each node in a node class, TCT, issue the following command:

```
mmcloudgateway service version -N TCT
```

The system displays output similar to this:

| Cluster | minReleaseLevel:                   | 5.0.1.0   |          |             |                    |
|---------|------------------------------------|-----------|----------|-------------|--------------------|
| Node    | Daemon                             | node name | TCT Type | TCT Version | Equivalent Product |
| Version |                                    |           |          |             |                    |
| 1       | jupiter-vm1192.pok.stglabs.ibm.com |           | Server   | 1.1.5       | 5.0.1.0            |

To display the client version of each node, issue the following command on the client node:

```
mmcloudgateway service version
```

The system displays output similar to this:

| Cluster | minReleaseLevel:                  | 5.0.1.0   |          |             |                    |
|---------|-----------------------------------|-----------|----------|-------------|--------------------|
| Node    | Daemon                            | node name | TCT Type | TCT Version | Equivalent Product |
| Version |                                   |           |          |             |                    |
| 4       | jupiter-vm649.pok.stglabs.ibm.com |           | Client   | 1.1.5       | 5.0.1.0            |

To verify the client version of a particular node, issue the following command:

```
mmcloudgateway service version -N jupiter-vm717
```

The system displays output similar to this:

| Cluster | minReleaseLevel: | 5.0.1.0   |          |             |                    |
|---------|------------------|-----------|----------|-------------|--------------------|
| Node    | Daemon           | node name | TCT Type | TCT Version | Equivalent Product |
| Version |                  |           |          |             |                    |
|         |                  |           |          |             |                    |

```
3 jupiter-vm717.pok.stglabs.ibm.com Client 1.1.5 5.0.1.0
```

To check for all nodes in a node class, issue the following command:

```
mmcloudgateway service version -N tct
```

The system displays output similar to this:

```
Cluster minReleaseLevel: 5.0.1.0
Node Daemon node name TCT Type TCT Version Equivalent Product Version

2 jupiter-vm482.pok.stglabs.ibm.com Server 1.1.5 5.0.1.0
1 jupiter-vm597.pok.stglabs.ibm.com Server 1.1.5 5.0.1.0
```

## Known limitations of Cloud services

This topic describes the limitations that are identified for Cloud services.

### **mmcloudgateway files migrate \* on a parent folder does not move all files within the subfolders**

Running the **mmcloudgateway files migrate** command to migrate all files (including the files within the subfolders) does not migrate all files within subfolders. It migrates only the leaf files within the current folder, from which the migrate command is issued. The migrate process skips the subfolders, by displaying the following warning message:

```
MCSTG00051E: File is not a regular file. Migration requests only support regular files.
error processing /<file-system-mount>/<folder1>/<folder-2>....
```

To migrate all files (including files within the subfolders) in one go, issue this command:

```
find <gpfs-mountpoint-folder-or-subfolder> -type f -exec mmcloudgateway files migrate {} +
```

This command passes the entire list of files to a single migrate process in the background as follows:

```
mmcloudgateway files migrate <file1> <file2> <sub-folder1/file1> <sub-folder2/file1>
```

### **Migrating Transparent cloud tiering specific configuration to cloud storage might lead to issues**

While you move data to an external cloud storage tier, it is required not to migrate files within the Transparent cloud tiering internal folder (.mcstore folder within the configured GPFS file system) to cloud storage. It might lead to undesirable behavior for the Transparent cloud tiering service. To address this issue, include the EXCLUDE directive in the migration policy.

Refer to the /opt/ibm/MCStore/samples folder to view sample policies that can be customized as per your environment and applied on the file system that is managed by Transparent cloud tiering.

### **Running mmcloudgateway files delete on multiple files**

Trying to remove multiple files in one go with the **mmcloudgateway files delete delete-local-file** command fails with a NullPointerException. This happens while you clean up the cloud metrics. Issue this command to remove the cloud objects:

```
find <gpfs-file-system> -type f -exec mmcloudgateway files delete {} \;
```

### **Range reads from the Cloud Object Storage is not supported for transparent recall.**

When a file is transparently recalled, the file is entirely recalled.

### **Policy-based migrations**

Policy-based migrations should be started only from Transparent cloud tiering server nodes. Client nodes should be used only for manual migration.

### **File names with carriage returns or non-UTF-8 characters**

Transparent cloud tiering does not perform any migration or recall operation on files whose names include carriage returns or non-UTF-8 characters.

### **File systems mounted with the nodev option**

If a file system is mounted with the nodev option, then it cannot be mounted to a directory with an existing folder with the same name as the file system. Transparent cloud tiering is not supported in this situation.

### **Administrator cannot add a container pair set while managing a file system with 'automount' setting turned on.**

Make sure that automount setting is not turned on while a file system is in use with Transparent cloud tiering.

### **Files created through NFS clients when migrated to the cloud storage tier**

If caching is turned on the NFS clients (with the --noac option) while mounting the file system, files that are migrated to the cloud storage tier remain in the co-resident status, instead of the non-resident status.

### **Transparent cloud tiering configured with proxy servers**

IBM Security Key Lifecycle Manager does not work when Transparent cloud tiering is configured with proxy servers.

### **Swift Dynamic Large Objects**

Transparent cloud tiering supports Swift Dynamic Large Objects only.

### **No support for file systems earlier than 4.2.x**

Cloud services support IBM Storage Scale file systems versions 4.2.x and later only.

### **Running reconciliation during heavy writes and reads on the file system**

Reconciliation fails when it is run during heavy I/O operations on the file system.

For current limitations and restrictions, see [IBM Storage Scale FAQs](#).

For more information, see the topic *Interoperability of Transparent Cloud Tiering with other IBM Storage Scale features* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

# Chapter 53. Managing file audit logging

The following topics describe various ways to manage file audit logging in IBM Storage Scale.

For more information about file audit logging, see *Introduction to file audit logging in IBM Storage Scale: Concepts, Planning, and Installation Guide*.

## Managing the list of monitored events

Use this information to manage file audit logging events.

For more information about file audit logging events, see *File audit logging events' descriptions* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

- To list the events that are being monitored for a currently enabled file audit logging file system, run the following command:

```
mmaudit <device> list --events
```

- To change the monitored events, run the following command:

```
mmaudit <device> update --events <Event1,Event2,...>
```

- To change the monitored events back to **ALL** events, run the following command:

```
mmaudit <device> update --events ALL
```

For more information, see the *mmaudit command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

## Manage and list currently enabled audits of all types

The **mmaudit all list** command can be used to see what type of audit is enabled on a file system.

The following output is an example of the information that is available when you run the **mmaudit all list** command:

| Audit Device | Cluster ID           | Audit Fileset Name | Retention (Days) | Audit Type (Possible Filesets)     |
|--------------|----------------------|--------------------|------------------|------------------------------------|
| fs0          | 11430652110915196903 | john1              | 25               | FILESET<br>dep1,dep2,ind1,ind2     |
| fs1          | 11430652110915196903 | john2              | 75               | SKIPFILESET<br>dep1,dep2,ind1,ind2 |
| fs2          | 11430652110915196903 | john3              | 25               | FSYS                               |



# Chapter 54. RDMA tuning

Read about tuning RDMA attributes to avoid problems in configurations with InfiniBand.

See the following sections of this help topic:

[“Settings for IBM Storage Scale 5.0.x and later” on page 913](#)

[“Settings for IBM Storage Scale 4.2.3.x” on page 913](#)

[“Suggested CPU tuning for Sandy Bridge processors” on page 914](#)

## Settings for IBM Storage Scale 5.0.x and later

### Registering the page pool to InfiniBand

If the GPFS daemon cannot register the page pool to InfiniBand, it fails with the following mmfs log messages:

```
VERBS RDMA Shutdown because pagepool could not be registered to Infiniband.
VERBS RDMA Try increasing Infiniband device MTTs or reducing pagepool size.
```

To resolve this problem, try adjusting the following mlx4\_core module parameters for the Mellanox Translation Tables (MTTs). This adjustment does not apply to mlx5\_core parameters.

1. Set **log\_mtts\_per\_seg** to 0. This value is the recommended one.
2. Increase the value of **log\_num\_mtt**.

For more information see the following links:

[How to increase MTT size in Mellanox HCA at Mellanox Documentation](#).

### Enabling verbsRdmaSend

The **verbsRdmaSend** attribute of the **mmchconfig** command enables or disables the use of InfiniBand RDMA rather than TCP for most GPFS daemon-to-daemon communications. When the attribute is disabled, only data transfers between an NSD server and an NSD client are eligible for RDMA. When the attribute is enabled, the GPFS daemon uses InfiniBand RDMA connections for daemon-to-daemon communications only with nodes that are at IBM Storage Scale 5.0.0 or later. For more information, see *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

## Settings for IBM Storage Scale 4.2.3.x

### Registering the page pool to InfiniBand

Follow the instructions for registering the page pool to InfiniBand in “[Settings for IBM Storage Scale 5.0.x and later” on page 913](#) earlier in this topic.

### Enabling verbsRdmaSend

Read the discussion of setting **verbsRdmaSend** in “[Settings for IBM Storage Scale 5.0.x and later” on page 913](#) earlier in this topic. For 4.2.3.x, be aware of the following points:

- Do not enable **verbsRdmaSend** in clusters greater than 500 nodes.
- Disable **verbsRdmaSend** if either of the following types of error appears in the mmfs log:
  - Out of memory errors
  - InfiniBand error IBV\_WC\_RNR\_RETRY\_EXC\_ERR

### Setting scatterBufferSize in very large clusters (> 2100 nodes)

The **scatterBufferSize** attribute of the **mmchconfig** command has a default value of 32768, which provides good performance under most conditions. However, if the CPU use on the NSD I/O

servers is high and client I/O is lower than expected, increasing the value of **scatterBufferSize** might improve performance. Try the following settings:

- For Mellanox FDR 10 InfiniBand: 131072.
- For Mellanox FDR 14 InfiniBand: 262144.

This attribute is not described in regular IBM Storage Scale documentation.

#### **Setting verbsRdmaPerNode in large clusters (> 100 nodes)**

The **verbsRdmaPerNode** attribute of the **mmchconfig** command sets the maximum number of RDMA data transfer requests that can be active at the same time on a single node. The default value is 1000. If the cluster is large (more than 100 nodes) the suggested value is the same value that is set for the attribute **nsdMaxWorkerThreads**.

This attribute is supported only in IBM Storage Scale version 4.2.x. For more information, see *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide 4.2.3*.

#### **Suggested CPU tuning for Sandy Bridge processors**

For Intel Core Sandy Bridge processors, if RDMA performance is less than expected, ensure that the C-states that reduce CPU voltage are disabled on the affected nodes.

---

# Chapter 55. Configuring Mellanox Memory Translation Table (MTT) for GPFS RDMA VERBS Operation

You need to configure the Mellanox Memory Translation Table (MTT) with correct page pool size for GPFS RDMA or Mellanox InfiniBand RDMA (VERBS) operation .

## How GPFS pagepool size affects Mellanox InfiniBand RDMA (VERBS) configuration

Improperly configuring the Mellanox MTT can lead to the following problems:

- Excessive logging of RDMA-related errors in the IBM Storage Scale log file.
- Shutdown of the GPFS daemon due to memory limitations. This can result in the loss of NSD access if this occurs on an NSD server node.

For more information, see the topic *GPFS and Memory* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

## Mellanox Variables

The Mellanox mlx4\_core driver module has the following two parameters that control its MTT size and define the amount of memory that can be registered by the GPFS daemon. The parameters are **log\_num\_mtt** and **log\_mtts\_per\_seg** and they are defined as a power of 2.

- **log\_num\_mtt** defines the number of translation segments that are used.
- **log\_mtts\_per\_seg** defines the number of entries per translation segment.

Each **log\_mtts\_per\_seg** maps a single page, as defined by the hardware architecture, to the mlx4\_core driver. For example, setting the variable **log\_num\_mtt** to 20 results in a value of 1,048,576 (segments) which is 2 to the power of 20. Setting the variable **log\_mtts\_per\_seg** to 3 results in the value of 8 (entries per segment) which is 2 to the power of 3. These parameters are set in the mlx4\_core module of the /etc/modprobe.conf file, or on a line at the end of /etc/modprobe.d/mlx4\_core.conf file, depending on your version of Linux. Here is an example of how the parameters can be set in those files.

Options **mlx4\_core log\_num\_mtt=23 log\_mtts\_per\_seg=0**

To check the configuration of the mlx4 driver use the following command:

```
cat /sys/module/mlx4_core/parameters/log_num_mtt
23
cat /sys/module/mlx4_core/parameters/log_mtts_per_seg
0
```

## GPFS pagepool mapping

When the GPFS daemon starts, and the **verbsRdma** parameter is enabled, GPFS attempts to register the pagepool with the mlx4\_core driver. Because the GPFS registers the pagepool twice, the values of the Mellanox parameters must allow the mapping memory to be at least twice the size of the GPFS pagepool. If the pagepool size is not a power of 2, the size is rounded up to the next power of 2 size. This rounded up size is used when registering the pagepool with the mlx4\_core driver. If the attempt to map the GPFS pagepool to the mlx4\_core driver fails the GPFS daemon will shut down and log messages similar to these.

```
VERBS RDMA Shutdown because pagepool could not be registered to Infiniband.
VERBS RDMA Try increasing Infiniband device MTTs or reducing pagepool size.
```

## Example to support GPFS pagepool of 32GB

If the GPFS pagepool is set to 32 GB, then the mapping of the RDMA for this pagepool must be at least 64 GB. In addition to the two Mellanox configuration variables described previously, you need to know the page size that is used by the architecture on which IBM Storage Scale is running.

**Note:** The x86 architecture uses a page size of 4096 bytes (4 K) and Power architecture (ppc64) uses a page size of 65536 (64 K). Here are the mappings for each architecture for GPFS pagepool of 32 GB.

**x86:**

```
log_num_mtt=24
log_mtts_per_seg=0
page size 4 K
2^log_num_mtt X 2^log_mtts_per_seg X page size
2^24 X 1 X 4096
16,777,216 X 1 X 4096 = 68,719,476,736 (64 GB)
```

**ppc64:**

```
log_num_mtt=20
log_mtts_per_seg=0
page size 64 K
2^log_num_mtt X 2^log_mtts_per_seg X page size
2^20 X 2^0 X 65,536
1,048,576 X 1 X 65,536 = 68,719,476,736 (64 GB)
```

# Chapter 56. Administering cloudkit

Refer to the following topics to assist you in administering **cloudkit**.

## Grant repository

Grants the repository access to the VPC.

After running the **cloudkit grant repository** command, the resources in the VPC can access the IBM Storage Scale rpms that are in the repository.

## Grant filesystem

Grants the filesystem access to the compute cluster.

After running the **cloudkit grant filesystem** command, the **cloudkit** remotely mounts the file system on the compute node.

## Mounting GPFS remote mount on compute nodes

You must explicitly mount a GPFS file system on a compute node by using the **cloudkit grant filesystem** command.

Before you run the **cloudkit grant filesystem** command, make sure that both storage and compute nodes were created in the IBM Storage Scale cloud cluster environment.

To mount a GPFS file system on compute nodes, enter the following command:

```
#./cloudkit grant filesystem
```

### Example:

```
./cloudkit grant filesystem
I: Logging at /root/scale-cloudkit/logs/cloudkit-26-7-2024_8-52-54.log
? Compute cluster type: IBM-Storage-Scale-Compute-Cluster
? Are you operating cloudkit in air-gapped or no-internet environment: No
? Select a storage cluster: ibm-stroage-scale | None | us-central1 | GCP
? Storage cluster management GUI username: administrator
? Storage cluster management GUI password: [? for help] *****
? Select a compute cluster (cluster that needs access to filesystem): ibm-compute-scale | None
| us-central1 | GCP
? Select filesystem mount point: /gpfs/fs1
? Compute cluster management GUI username: administrator
? Compute cluster management GUI password: [? for help] *****
? Remote filesystem mount point: /ibm/remotefs
I: Obtaining IBM Storage Scale storage cluster definition.
I: Obtaining IBM Storage Scale compute cluster definition.
I: Updating compute security group 'ibm-compute-scale-cluster-tag' to allow traffic from
storage security group 'ibm-stroage-scale-cluster-tag'.
I: Updating storage security group 'ibm-stroage-scale-cluster-tag' to allow traffic from
compute security group 'ibm-compute-scale-cluster-tag'.
I: To grant access to "ibm-stroage-scale" filesystem again in the future, you can run:
 cloudkit grant filesystem --airgap=false --region us-central1 --cloud-platform GCP --
resource-name ibm-stroage-scale-ibm-compute-scale --resource-group-ref None --remotefs-mount-
point /ibm/remotefs --storage-cluster-name ibm-stroage-scale --strg-gui-username administrator
--compute-cluster-mode IBM-Storage-Scale-Compute-Cluster --compute-cluster-name ibm-compute-
scale --cmp-gui-username administrator --jumphost-login-user gcpuser --jumphost-asg-ref https://
www.googleapis.com/compute/v1/projects/spectrum-XXXX-XXXXX/zones/us-central1-c/
instanceGroupManagers/ibm-stroage-scale-bastion-asg --jumphost-private-key-path /root/scale-
cloudkit/data/gcp/ibm-stroage-scale/bastion_ssh_keys/id_rsa --jumphost-sec-ref ibm-stroage-
scale-bastion-tag --project-id spectrum-XXXX-XXXXX --credential-jsonpath /root/.config/gcloud/
legacy_credentials/test@spectrum-XXX-XXXXX.iam.gserviceaccount.com/adc.json

I: Initiating remote mount configuration.
100% |
```

  
[ (10/10, 3 it/min)

```
I: IBM Storage Scale cluster configuration completed.
I: Access to IBM Storage Scale storage cluster 'ibm-stroage-scale' has been granted for compute
cluster 'ibm-compute-scale'.
```

## Unmounting GPFS remote mount on compute nodes

You can unmount a GPFS file system by using the **cloudkit revoke filesystem** command.

To perform some GPFS administration tasks, you must first unmount the file system.

To unmount a GPFS file system by using the **revoke** command, enter the following line:

```
#./cloudkit revoke filesystem
```

### Example:

```
./cloudkit revoke filesystem
I: Logging at /root/scale-cloudkit/logs/cloudkit-26-7-2024_9-6-21.log
=====
| Note !
=====
| Revoke cluster involves unmount of filesystem. Make sure the I/O is
| stopped and data is flushed before proceeding further.
=====
? Are you operating cloudkit in air-gapped or no-internet environment: No
? Select a remotemount: ibm-storage-scale | None | us-central1 | GCP | ibm-compute-scale |
None | us-central1 | GCP | /gpfs/fs1
? Storage cluster management GUI username: administrator
? Storage cluster management GUI password: [? for help] *****
? Compute cluster management GUI username: administrator
? Compute cluster management GUI password: [? for help] *****
I: To revoke access to "ibm-storage-scale" filesystem again in the future, you can run:
 cloudkit revoke filesystem --airgap=false --region us-central1 --cloud-platform GCP --
resource-group-ref None --remoteefs-mount-point /ibm/remoteefs --storage-cluster-name ibm-storage-
scale --strg-gui-username administrator --cmp-gui-username administrator --jumphost-login-user
gcpuser --jumphost-asg-ref https://www.googleapis.com/compute/v1/projects/spectrum-xxxxx-xxxxx/
zones/us-central1-c/instanceGroupManagers/ibm-storage-scale-bastion-asg --jumphost-private-key-
path /root/scale-cloudkit/data/gcp/ibm-storage-scale/bastion_ssh_keys/id_rsa --jumphost-sec-ref
ibm-storage-scale-bastion-tag --project-id spectrum-xxxx-xxxx --credential-jsonpath /
root/.config/gcloud/legacy_credentials/prvndev@spectrum-xxxxx-xxxxx.iam.gserviceaccount.com/
adc.json

I: Obtaining IBM Storage Scale storage cluster definition.
I: Obtaining IBM Storage Scale compute cluster definition.
I: Initiating remote mount revoke configuration.
100% |

=====
| (10/10, 8 it/min)
I: IBM Storage Scale cluster configuration completed.
I: Updating compute security group 'ibm-compute-scale-cluster-tag' to revoke traffic from
storage security group 'ibm-storage-scale-cluster-tag'.
I: Updating storage security group 'ibm-storage-scale-cluster-tag' to revoke traffic from
compute security group 'ibm-compute-scale-cluster-tag'.
I: Access to IBM Storage Scale cluster 'ibm-compute-scale' has been revoked.
```

## Editing or scaling out an IBM Storage Scale cloud cluster

Scale out an IBM Storage Scale cloud cluster by using the **cloudkit edit cluster** command.

The **cloudkit edit cluster** command helps to increase the filesystem capacity of an existing IBM Storage Scale cloud cluster by performing:

- Expansion of storage cluster instances
- Expansion of compute cluster instances

Table 74. Instance maximum limit

| Deployment profiles                       | Compute node limit | Storage node limit |
|-------------------------------------------|--------------------|--------------------|
| Throughput-Performance-Persistent-Storage | 64                 | 64                 |
| Throughput-Performance-Scratch-Storage    | 64                 | 64                 |

Table 74. Instance maximum limit (continued)

| Deployment profiles                   | Compute node limit | Storage node limit |
|---------------------------------------|--------------------|--------------------|
| Throughput-Advance-Persistent-Storage | 64                 | 10                 |
| Balanced                              | 64                 | 64                 |

To expand a cluster, enter the following command:

```
#./cloudkit edit cluster
```

**Example:**

```
./cloudkit edit cluster

I: Logging at /root/scale-cloudkit/logs/cloudkit-26-7-2024_9-12-34.log
? Are you operating cloudkit in air-gapped or no-internet environment: No
? Select a cluster: ibm-storage-scale | None | us-central1 | GCP
? Select the role: Storage-only
I: Opted cluster 'ibm-storage-scale' is a 'Storage-only' cluster.
I: Opted cluster 'ibm-storage-scale' currently has '4' nodes.
? Storage node count: 6
I: Requested to expand the cluster by '2' node(s).
? Do you wish to view cost before provisioning the resources (this requires infracost
api_key): No
I: To edit this cluster again in the future, you can run:
 cloudkit edit cluster --airgap=false --region us-central1 --cloud-platform GCP --
resource-name ibm-storage-scale --resource-group-ref None --view-cost=false --jumphost-login-
user gcpuser --jumphost-asg-ref https://www.googleapis.com/compute/v1/projects/spectrum-
scale-349401/zones/us-central1-c/instanceGroupManagers/ibm-storage-scale-bastion-asg --jumphost-
private-key-path /root/scale-cloudkit/data/gcp/ibm-storage-scale/bastion_ssh_keys/id_rsa --
jumphost-sec-ref ibm-storage-scale-bastion-tag --project-id xxxxxxx-xxxx-xxxx-xxxx --credential-
jsonpath /root/.config/gcloud/legacy_credentials/xxxxx@xxxxx-xxxx-xxxxx.iam.gserviceaccount.com/
adc.json --strg-node-count 6 --role Storage-only

I: Past state of resources exists. Reusing it.
I: Initiating resource provisioning actions proposed in the scale instances IaC plan.
100% |

[Progress bar] (10/10, 20 it/min)

I: Cluster resource provisioning actions proposed in the plan completed.

I: Waiting for instances to obtain running state.
I: Initiating cluster configuration based on the provisioned resource inventory.
100% |

[Progress bar] (10/10, 2 it/min)

I: IBM Storage Scale cluster configuration completed.
I: IBM Storage Scale cluster 'ibm-storage-scale' has been edited successfully.
```

**Note:** NSD disks are not balanced after an edit operation. If you want to rebalance the existing file system data across disks, run the **mmrestripefs** command.

## Enabling IBM Storage Scale GUI access by using JumpHost

The IBM Storage Scale GUI can be accessed for an IBM Storage Scale cloud cluster by using the **cloudkit port-forward** command.

The **cloudkit port-forward** command opens the corresponding GUI ports in the IBM Storage Scale cloud cluster to access the IBM Storage Scale GUI.

To grant GUI access, enter the following command:

```
#./cloudkit port-forward
```

### Example:

```
#!/cloudkit port-forward
I: Logging at /root/scale-cloudkit/logs/cloudkit-20-9-2023_0-54-41.log
? Cloud platform name: GCP
? Cluster name: ibm-strg-inst01
? Bastion/JumpHost instance login username: gcpuser
? Bastion/JumpHost instance public ip address: 1.1.1.1
? Bastion/JumpHost SSH private key file path (will be used only for configuration): /root/
scale-cloudkit/data/gcp/ibm-strg-inst01/bastion_ssh_keys/id_rsa
? Bastion CIDR allow list: 1.1.1.1/32
I: Adding firewall rule to allow GUI traffic from client cidr '[1.1.1.1/32]'.
I: Initiating GUI access configuration based on the provisioned resource inventory.
100% |
```

```
| (4/10, 14 it/min)
I: Access to GUI service of IBM Storage Scale cluster 'ibm-strg-inst01' has been granted on
Port '3126'.
I: IBM Storage Scale GUI:
 https://35.208.208.244:3126/
I: To use this cluster via REST interface, you can run (an example API):
 curl -u '<gui_username>:<gui_password>' -X GET -k https://35.208.208.244:3126/
scalemgmt/v2/info
[root@exampleserver cloudkit]
```

## Enabling and disabling repository access

Learn how to enable and disable **gpfs rpms** in a repository.

The **cloudkit** repository holds all the required gpfs rpms that are needed for installing an IBM Storage Scale cloud cluster.

To enable access to a repository, enter the following command:

```
#!/cloudkit grant repository
```

### Example:

```
#!/cloudkit grant repository
I: Logging at /root/scale-cloudkit/logs/cloudkit-20-9-2023_1-9-15.log
? Cloud platform name: GCP
? Select repository name: ibm-scale-repo
? Service account email: prvndev@spectrum-scale-349401.iam.gserviceaccount.com
I: Artifactory registry 'ibm-scale-repo-5-1-9-0-rhel-8' reader access binding with service
email 'exampleaccount@spectrum-scale-111111.iam.gserviceaccount.com' has been configured.
I: Artifactory registry 'ibm-scale-repo-5-1-9-0-rhel-9' reader access binding with service
email 'exampleaccount@spectrum-scale-111111.iam.gserviceaccount.com' has been configured.
I: Access to IBM Storage Scale repository 'ibm-scale-repo' has been granted.
```

To disable access to a repository, enter the following command:

```
#!/cloudkit revoke repository
```

### Example:

```
#!/cloudkit revoke repo
I: Logging at /root/scale-cloudkit/logs/cloudkit-20-9-2023_1-9-54.log
? Cloud platform name: GCP
? Select repository name: ibm-scale-repo
I: Revoking binding '[role:"roles/artifactregistry.reader"
members:"serviceAccount:exampleaccount@spectrum-scale-111111.iam.gserviceaccount.com"]' for
artifact registry 'ibm-scale-repo-5-1-9-0-rhel-8'.
I: Revoking binding '[role:"roles/artifactregistry.reader"
members:"serviceAccount:exampleaccount@spectrum-scale-111111.iam.gserviceaccount.com"]' for
artifact registry 'ibm-scale-repo-5-1-9-0-rhel-9'.
I: Access to IBM Storage Scale repository 'ibm-scale-repo' has been revoked.
```

## Enabling AFM caching

IBM Storage Scale active file management (AFM) to cloud storage enables automatic tiering of application data to Amazon Simple Storage Service (Amazon S3) or Google Cloud Storage (GCS).

For a detailed explanation of the AFM feature, see the *Introduction to AFM to cloud object storage* section in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.

If the cloud object storage target is AWS S3, the cloudkit simplifies the authentication setup that is needed by using the AWS instance IAM profile. Whereas if the target is GCS, the service account that is chosen during the configuration is used to generate the HMAC credentials, which are further used to setup authentication to the GCS bucket from gateway nodes.

To enable AFM to cloud object storage, use the **./cloudkit caching setup** command. The following steps illustrate a simplified setup of AFM caching.

1. Deploy the AFM gateway nodes.

Use the **./cloudkit edit cluster** command to provision or add AFM gateway nodes to an existing storage cluster or combined cluster.

**Example:**

```
./cloudkit edit cluster
I: Logging at /root/scale-cloudkit/logs/cloudkit-23-7-2024_6-28-28.log
? Are you operating cloudkit in air-gapped or no-internet environment: No
? Select a cluster: scale-demo-cls | None | us-central1 | GCP
? Select the role: Gateway-only
I: Opted cluster 'scale-demo-cls' currently has '0' gateway nodes.
? IBM Storage Scale Active File Management (AFM) Gateway instance type: n1-standard-2 |
vCPU(2) | RAM (7.50 GB) | Egress Network Bandwidth (Up to 10 Gbps)
? IBM Storage Scale Active File Management (AFM) Gateway node count: 1
I: Requested to expand the gateway nodes by '1' node(s).
? Do you wish to view cost before provisioning the resources (this requires infracost
api_key): (y/N)
```

2. Configure AFM to cloud object storage.

To enable an AFM relationship, issue the **./cloudkit caching setup** command.

**Example:**

```
]# ./cloudkit setup caching
I: Logging at /root/scale-cloudkit/logs/cloudkit-23-7-2024_6-50-1.log
? Are you operating cloudkit in air-gapped or no-internet environment: No
? Select a storage cluster: scale-demo-cls | None | us-central1 | GCP
? Storage cluster management GUI username: administrator
? Storage cluster management GUI password: [? for help] ****
? IBM Storage Scale Active File Management caching target type: Cloud-Object-Storage
? Fileset name: afmcos_fset
? IBM Storage Scale Active File Management caching mode: Single-writer | sw | One cache
fileset does all the writing
? AFM caching target name: Select this option for custom input.
? AFM Cloud Object Storage (bucket) name: scale-objectstore
I: Bucket 'scale-objectstore' has been created.
? Select filesystem mount point: /gpfs/fs1
```



# Chapter 57. Administering AFM

The following topics assist you in Administering AFM.

## Creating an AFM relationship by using the NFS protocol

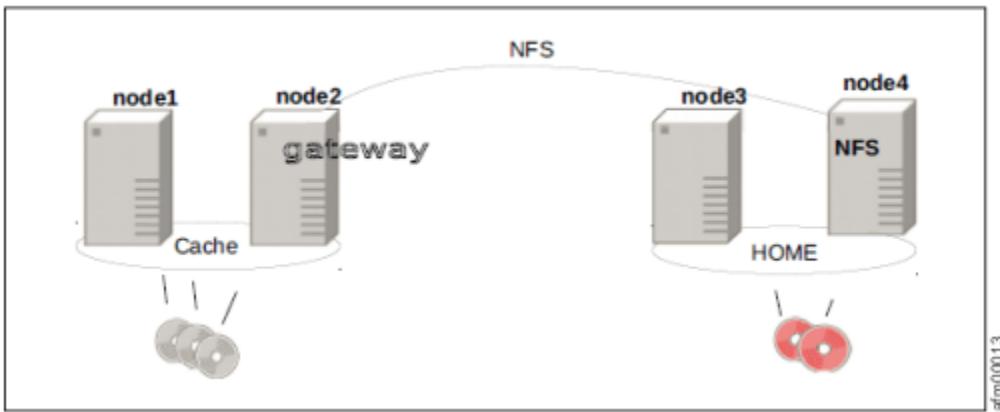


Figure 33. A demonstration setup of an AFM relationship

**Note:** NFS server maintains files in cache for 90 seconds after replication on AFM home. If data of an AFM home using CES NFS is exported via SMB, SMB clients accessing those files during that period can experience sharing violations.

### Setting up the home cluster

Use this procedure to set up the home cluster.

1. Set up a home cluster. For more information, see *Creating your GPFS cluster in IBM Storage Scale: Administration Guide*.
2. Configure a cache relationship for AFM filesets by using the home cluster that you created.

The relationship between the home cluster and the cache cluster is set up by using NFS exports that are defined on home cluster. The home cluster exports NFS mount points that AFM cache cluster uses to synchronize data.

3. Create a file system and mount this file system on the home cluster. For more information, see *Managing file systems in IBM Storage Scale: Administration Guide*.

The home side of the relationship is defined at the NFS share level. The home contains all the data available from the NFS mount point.

4. At the home cluster, create and link one or more filesets. For more information, see *Filesets in IBM Storage Scale: Administration Guide*.

These filesets are used to set up NFS exports at the home cluster. These export paths are fileset junction paths where filesets are linked.

5. Export the fileset, one fileset at a time. Do one of the following steps:

**Note:** In this example, the IP address of the Gateway node of the cache cluster is 192.168.1.2. As an Administrator, ensure that NFS exports are accessible only to nodes at the cache.

- If you are using KNFS, complete the following steps for KNFS export of the home file system:

- a. Add fileset junction path to the `/etc/exports` file of home cluster export servers. Exported path must have the necessary permissions. **no\_root\_squash** and **rw** are mandatory and **fsid** is optional. An example of an NFS export entry with fsid at home side is as follows:

```
/gpfs/fs1/sw2
192.168.1.2(rw,nohide,insecure,no_subtree_check,sync,no_root_squash,fsid=1069)
```

- b. Restart the NFS services on home cluster export servers. Because you restarted the NF services, all the necessary NFS services start and the specified path to be used by the AFM cache is exported.
- c. If the file system goes down, you must export the file system again. The gateway nodes at the cache site must have access to the exported directory and can be mounted by using NFS.
- If you want to configure a file system on the home cluster with protocols nodes, complete the following steps:
  - a. Use the **mmnfs export add** command to create export on junction path or the gpfs path of user choice.

```
mmnfs export add /ibm/gpfs0/nfsexport --client \
 "192.168.1.2(Access_Type=RW,Squash=no_root_squash,SecType=sys)"
```

Use the **mmnfs export list** command to list NFS exports:

```
mmnfs export list
```

A sample output is as follows:

| Path                 | Delegations | Clients |
|----------------------|-------------|---------|
| /ibm/gpfs0/nfsexport | none        | *       |

```

```

```
/ibm/gpfs0/nfsexport none *
```

Do not edit `/etc/exports` or any other NFS configuration files manually, and do not restart NFS services after the export is created.

6. After you export filesets at the home cluster, run **mmafmconfig enable /ibm/gpfs0/nfsexport**. For more information about the command, see *mmafmconfig* in *IBM Storage Scale: Command and Programming Reference Guide*.

**Note:**

- a. Ensure that you add the IP addresses of all gateway nodes of the cache cluster. Multiple IP addresses can be indicated by a comma-separated list. Update the list of IP addresses whenever you add or remove a gateway node.
- b. Ensure that the KNFS or NFS server at home is restarted, and home exports are available. CES NFS does not require a restart.
- c. If both NFS and GPFS start automatically at the start-up time, ensure that GPFS starts before NFS as NFS can export GPFS only if it is loaded. If NFS starts before GPFS, run **exportfs -r**.
- d. If an Ubuntu server is used as an AFM DR secondary server, configure the AFM secondary server so that the NFS server service `rpc.mountd` does not start with `--manage-gids`. `rpc.mountd --manage-gids` is not applicable for CES NFS or Native GPFS protocol.
- e. When an AFM-DR secondary fileset is exported by using an NFS server, set the `manage_gids=False` option. The server does not start because of this configuration. To change the option value, see [“Making bulk changes to NFS exports” on page 353](#).

This option is not applicable when an AFM-DR secondary fileset is connected over the native GPFS protocol.

## Setting up the cache cluster

Use this procedure to set up the cache cluster.

1. To identify the nodes of the GPFS cluster that function as the application nodes and the nodes that function as gateway nodes, run the following command:

```
mmclscluster
```

2. After you identify the nodes, run the following command to assign the role of a gateway node to the identified nodes:

```
mmchnode --gateway -N Node1,Node2,...
```

3. To ensure that GPFS started, run the following command:

```
mmgetstate -a
```

4. To mount the file system, run the following command:

```
mmmount filesystemname
```

5. To create an AFM fileset and link the fileset, run the following command:

```
mmcrlfileset filesystemname Fileset -p afmTarget=Home:Home-Exported-Path
--inode-space=new -p afmMode=single-writer | read-only | local-updates | independent-writer
```

```
mmlinkfileset filesystemname fileset -J /filesystem-path/fileset
```

6. Access the AFM fileset.

To check the fileset state and other details, run the following command:

```
mmafmctl filesystemname getstate
```

## Example of creating an AFM relationship by using the NFS protocol

You can use this example to create an AFM relationship between the home cluster and the cache cluster by using the NFS protocol.

In this example, the fs1 file system is mounted on the /gpfs/fs1 path at the cache cluster and the home cluster. A single-writer (SW) fileset is created at the cache cluster and data is synched to the home fileset and the files are verified. Similarly, other AFM modes fileset can be created by using this example.

1. Set up the home cluster.

You can configure the home exports by using either CES NFS or the default NFS, which is available with the operating system.

- a. To create a fileset at a home cluster, run the following command:

```
mmcrlfileset fs1 fset001 --inode-space new --inode-limit 1000000
```

- b. To link the fileset, run the following command:

```
mmlinkfileset fs1 fset001 -J /gpfs/fs1/fset001
```

- c. To enable AFM support for extended attributes and sparse files, configure the created fileset.

- i) If you use CES NFS, run the following command to export the path:

```
mnfs export add /gpfs/fs1/fset001 -c "<client Nodes IP/
range>(Access_Type=R0,Squash=no_root_squash)"
```

For more information about the **mnfs** command, see *mnfs in IBM Storage Scale: Command and Programming Reference Guide*

ii) If you use the default NFS, which is available with the operating system, do the following steps:

Update /etc/exports and add the following entry:

```
/gpfs/fs1/fset001 <client Nodes IP/range>(rw,no_root_squash,no_subtree_check,fsid=101)
```

Restart the NFS server.

```
exportfs -ra
```

Verify that the added export is shown properly.

```
showmount -e | grep fset001
/gpfs/fs1/fset001
```

**Note:** The client nodes IP/range must be the gateway node at the cache cluster.

2. Do the following steps at the cache cluster:

a. Identify a node and designate the node as a gateway node.

b. To provide a gateway role to the node in the cache cluster, run the following command:

```
mmchnode --gateway -N <Node>
```

The system displays a similar output as follows:

```
Wed Oct 8 22:35:42 CEST 2019: mmchnode: Processing node <Node>
mmchnode: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

c. Create an AFM fileset.

```
mmcrfileset fs1 fileset_SW -p afmtarget=<home export server>:/gpfs/fs1/fset001 -p
afemode=single-writer --inode-space new
```

The system displays a similar output as follows:

```
Fileset fileset_SW created with id 1 root inode 131075.
```

d. Link the AFM fileset at the cache cluster to sync the AMF relationship with the home cluster.

```
mmmlinkfileset fs1 fileset_SW -J /gpfs/fs1/fileset_SW
```

The system displays a similar output as follows:

```
Fileset fileset_SW linked at /gpfs/fs1/fileset_SW
```

e. Check whether the state of the cache fileset is inactive.

```
mmafmctl fs1 getstate
```

The system displays a similar output as follows:

| Fileset Name | Fileset Target               | Cache State | Gateway Node | Queue Length | Queue numExec |
|--------------|------------------------------|-------------|--------------|--------------|---------------|
| fileset_SW   | nfs://node4/gpfs/fs1/fset001 | Inactive    |              |              |               |

**Note:** Ensure that the home export is mountable on the gateway node at the cache cluster. AFM internally mounts the home NFS export by using NFS v3.

f. Create a test file 'a' to move the AFM SW fileset from the inactive state to the active state.

```
touch /gpfs/fs1/fileset_SW/a
ls -l /gpfs/fs1/fileset_SW/
```

The system displays a similar output as follows:

```
total 1
drwx----- 4 root root 4096 Oct 8 20:38 a
```

The AFM fileset state changes to the active state after some time.

The active state indicates that the home and cache relationship is established and synced.

```
mmafmctl fs1 getstate
```

The system displays a similar output as follows:

| Fileset Name | Fileset Target               | Cache State | Gateway Node | Queue Length | Queue numExec |
|--------------|------------------------------|-------------|--------------|--------------|---------------|
| fileset_SW   | nfs://node4/gpfs/fs1/fset001 | Active      | node2        | 0            | 5             |

Here, 'node2' is the gateway node at the cache cluster and 'node4' is the home export server.

g. Create more test files in the cache fileset and verify synchronization at the home cluster.

```
cd /gpfs/fs1/fileset_SW
for i in 1 2 3 4 ; do date > file$i; done
```

i) List the created files in the cache fileset.

```
ls -l
```

The system displays a similar output as follows:

```
total 3
-rw-r--r-- 1 root root 30 Oct 9 20:22 a
-rw-r--r-- 1 root root 30 Oct 9 20:25 file1
-rw-r--r-- 1 root root 30 Oct 9 20:25 file2
-rw-r--r-- 1 root root 30 Oct 9 20:25 file3
-rw-r--r-- 1 root root 30 Oct 9 20:25 file4
```

h. Check the state of the AFM fileset.

```
mmafmctl fs1 getstate
```

The system displays a similar output as follows:

| Fileset Name | Fileset Target               | Cache State | Gateway Node | Queue Length | Queue numExec |
|--------------|------------------------------|-------------|--------------|--------------|---------------|
| fileset_SW   | nfs://node4/gpfs/fs1/fset001 | Dirty       | node2        | 8            | 5             |

i. Wait for sometime and check again if the fileset state is 'Active'.

```
mmafmctl fs1 getstate
```

The system displays a similar output as follows:

| Fileset Name | Fileset Target               | Cache State | Gateway Node | Queue Length | Queue numExec |
|--------------|------------------------------|-------------|--------------|--------------|---------------|
| fileset_SW   | nfs://node4/gpfs/fs1/fset001 | Active      | node2        | 0            | 13            |

3. To verify the available files in the home cluster, run the following command:

```
ls -l /gpfs/fs1/fset001
```

The system displays a similar output as follows:

```
total 3
-rw-r--r-- 1 root root 30 Oct 9 20:25 a
-rw-r--r-- 1 root root 30 Oct 9 20:28 file1
-rw-r--r-- 1 root root 30 Oct 9 20:28 file2
-rw-r--r-- 1 root root 30 Oct 9 20:28 file3
-rw-r--r-- 1 root root 30 Oct 9 20:28 file4
```

## Example of AFM support for Kerberos-enabled NFS protocol exports

AFM supports CES-enabled NFS protocol services and configuration without CES NFS (NFS shipped with OS)

AFM can use support of file access protocol authentication by using AD & LDAP setup. Kerberos can be used to add higher security for data transfer between Home/Secondary & Cache/Primary servers. As an administrator, you can configure file access for protocol authentication along with Kerberos configuration for client access. AFM & AFM-DR require all gateway nodes at the cache or primary cluster to be configured and authenticated as a Kerberos client. Home/Secondary cluster must be configured to export Kerberos-enabled NFS mounts.

Complete the following steps:

1. Ensure that the AFM home or secondary cluster is NFS service-enabled - Cluster Export Services (CES) protocol services or default NFS service.
2. Configure authentication and ID mapping for file-access by using LDAP or AD because it is required for Kerberos-enabled exports at AFM home or secondary fileset.
3. At cache or primary clusters, all gateway nodes must be configured and authenticated as a Kerberos client to enable mounting Kerberos-enabled NFS exports at cache or primary. Run the following command to export NFS Target mount paths in order of security levels like sys, krb5, krb5i, or krb5p.

```
mmnfs export add <Target_Path_Home> -c '[GatewayIPAddresses|*]
(Access_Type=RW,Squash=no_root_squash,SECTYPE=krb5i)'
```

4. Run the following command to enable clients to mount above export paths at gateway nodes by using NFS V3.

```
#mount -t nfs -o vers=3,sec=krb5i <Home>/<Target_Path_Home> /mnt1
```

5. Enable **afmEnableNFSSec** at cache or primary cluster to yes. Run the following command:

```
#mmchconfig afmEnableNFSSec=yes -i
```

6. Create an AFM fileset for the prepared target and link the fileset to the target. Run the following commands:

```
mmcrfileset filesystemname Fileset -p afmTarget=Home:/<Target_Path_Home>
--inode-space=new -p afmMode=single-writer | read-only | local-updates | independent-writer
```

```
#mmmlinkfileset filesystemname fileset -J /filesystem-path/fileset
```

7. You can access this fileset:

```
#mmafmctl <fs name> getstate -j <Fileset>
```

## Creating an AFM relationship by using GPFS protocol

The following topics describe how to set up the home and cache cluster.

### Setting up the home cluster

This topic lists the steps to set up the home cluster.

1. Create a home cluster. For more information, see *Creating your GPFS cluster in IBM Storage Scale: Administration Guide*.
2. Create a file system on the created home cluster. For more information, see *Managing file systems in IBM Storage Scale: Administration Guide*.
3. Create filesets on the home cluster. For more information, see *Filesets in IBM Storage Scale: Administration Guide*.

4. Enable remote access to the created file system on the home cluster. For more information, see *Accessing a remote GPFS file system in IBM Storage Scale: Administration Guide*.
5. To configure the exported path on the home cluster for AFM, run **mmafmconfig enable /ibm/gpfs0/nfsexport** on the home cluster.
6. If encryption is configured on a home site that is running on a file system, which AFM uses as a GPFS backend target (multi-cluster remote mount), of an IBM Storage Scale cluster, ensure that the cache cluster is also configured the same way as the home cluster. Because of this configuration, AFM can access the files on the target file system for the replication.

The file system with its filesets, is now accessible to the AFM cache cluster.

## Setting up the cache cluster

This topic lists the steps to set up the cache cluster.

1. To determine the nodes of the GPFS cluster that functions as application nodes and the nodes that function as the gateway nodes, run the **mmchnode --gateway -N Node1,Node2,...** command.
2. To start GPFS, run the **mmstartup -a** command.
3. To mount the file system, run the **mmmount Device -a** command.
4. Mount the home filesystem on the cache cluster remotely.
5. To create an AFM fileset and link it, run the following command:

```
mmcrfileset Device Fileset -p afmTarget=Home-Path --inode-space=new
-p afmMode=single-writer | read-only | local-updates | independent-writer
mmlinkfileset device fileset -J /fsmount-path/fileset
```

6. Access the AFM fileset.

**mmafmctl Device getstate** run on the cache cluster displays the fileset state and other details.

## Example of creating an AFM relationship by using the GPFS protocol

How to create an AFM relationship between the home cluster and cache cluster by using the GPFS protocol.

#designate a node as gateway

```
node1:~ # mmchnode --gateway -N node2
```

```
Wed Oct 8 22:35:42 CEST 2014: mmchnode: Processing node node2.site
mmchnode: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

# Remote mount the home filesystem on the cache

```
node1:/var/mmfsssl # mmremoteefs show all
```

| Local Name | Remote Name | Cluster name | Mount Point    | Mount Options | Automount | Drive | Priority |
|------------|-------------|--------------|----------------|---------------|-----------|-------|----------|
| remoteHOME | fshome      | node3.home   | /remote/fshome | rw            | no        | -     | 0        |

```
node1:/var/mmfsssl # mmmount remoteHOME -a
```

```
Thu Oct 23 23:28:32 CEST 2014: mmmount: Mounting file systems ...
```

#Create AFM with GPFS protocol as the transport layer

```
node1:/var/mmfsssl # mmcrfileset fs1 fileset_IW -p afmtarget=gpfs:///remote/
fshome -p afemode=iw --inode-space=new
```

```
Fileset fileset_IW created with id 1 root inode 131075.
```

```
node1:/var/mmfs/ssl # cd /gpfs/cache
node1:/gpfs/cache # mmmlinkfileset fs1 fileset_IW
```

```
Fileset fileset_IW linked at /gpfs/cache/fileset_IW
```

```
node1:/gpfs/cache # mmafmctl fs1 getstate
```

| Fileset    | Name | Fileset              | Target | Cache | State    | Gateway | Node | Queue | Length | Queue | numExec |
|------------|------|----------------------|--------|-------|----------|---------|------|-------|--------|-------|---------|
| fileset_IW |      | gpfs://remote/fshome |        |       | Inactive |         |      |       |        |       |         |

```
node1:/gpfs/cache # cd fileset_IW
```

```
node1:/gpfs/cache/fileset_IW # ll
total 97
drwx----- 65535 root root 32768 Oct 23 23:47 .afm
drwx----- 65535 root root 32768 Oct 23 23:47 .pconflicts
drwx----- 65535 root root 32768 Oct 23 23:47 .ptrash
dr-xr-xr-x 2 root root 32768 Jan 1 1970 .snapshots
```

```
node1:/gpfs/cache/fileset_IW #mmmount remoteHOME -a
access the fileset / wait 60 seconds
node1:/gpfs/cache/fileset_IW # mmafmctl fs1 getstate
Fileset Name Fileset Target Cache State Gateway Node Queue Length Queue numExec

fileset_IW gpfs://remote/fshome Active node1 0 13
```

## Checking the synchronization status of an AFM fileset

During the data synchronization, check the status of an AFM fileset by using the **mmafmctl** command.

1. Check the synchronization status from an AFM fileset to the home.

```
mmafmctl fs2 getstate -j fileset-RO --write-stats
```

A sample output is as follows:

| Fileset    | Name | Total Written Data<br>(Bytes) | N/w Throughput<br>(KB/s) | Total Pending Data<br>to Write(Bytes) | Estimated<br>Completion time |
|------------|------|-------------------------------|--------------------------|---------------------------------------|------------------------------|
| fileset-RO |      | 8133222872                    | 84888                    | 60648358095                           | 11 (Min)                     |

2. Check the synchronization status from the home to an AFM cache fileset.

```
mmafmctl fs1 getstate -j fileset-IW --read-stats
```

A sample output is as follows:

| Fileset    | Name | Total Read Data<br>(Bytes) | N/w Throughput<br>(KB/s) | Total Pending Data<br>to Write(Bytes) | Estimated<br>Completion time |
|------------|------|----------------------------|--------------------------|---------------------------------------|------------------------------|
| fileset-IW |      | 79585873                   | 8635                     | 2459959296                            | 4 (Min)                      |

### Note:

- If the **afmFastCreate** parameter value is set to yes or AFM to cloud object storage is enabled on a fileset, the **--read-stats** and **--write-stats** options show information such as N/w Throughput, Total Pending Data, Estimated Completion time only during the recovery or resync operation. During regular operations, the **--read-stats** or **--write-stats** option shows only Total Written Data.
- During recovery event, it might take some time for AFM to collect recovery data and queue operations to the AFM gateway node. The synchronization status is not shown until data is queued to the AFM gateway and the write operation is synchronized to the home.

For more information about these options, see *mmafmctl command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

## Pre-populating metadata by using the out of band prefetch

Sometimes, prefetching of data and metadata from a home target might take more time. During prefetch, AFM queues a lookup operation to a gateway node for each file. This operation might take several minutes or hours based on the amount of data and network latency. If metadata on the home is modified, AFM must re-populate new metadata on an AFM cache fileset.

To overcome this delay, AFM can pre-populate the metadata quickly on the cache fileset by using the metadata list-file. Pre-population by using the list-file does not queue any operation on a gateway node and quickly pre-populates the fileset locally.

The out of band population is a two-step process where metadata list-file needs to be generated on the home site by issuing the **mmafmctl getOutBandList** command. By using this generated metadata list-file, prefetch can be performed on the cache site by issuing the **mmafmctl --outband** command.

Consider the following points before pre-populating metadata:

- The out of band metadata pre-population can only be performed on an AFM fileset by using NSD backed targets. That is, multi-cluster backend. For more information about the NSD backend, see the *Backend protocol: NFS versus NSD* section in the *IBM Storage Scale: Administration Guide*.
- Both home and cache site must be running IBM Storage Scale 5.1.9 or higher.
- Pre-population of metadata is only supported for POSIX metadata only, such as UID, GID, file permissions, size, mtime. It does not support other metadata, such as NFS v4 acl, EA/xattr.
- If the home or source is not an IBM Storage Scale fileset but an IBM Storage Scale file system path, the fileset can be specified as 'root' when you are issuing the **prefetch** command.
- The prefetch can be run for deleting files with the --delete option first, and then run the --outband prefetch.

Run the following steps in a loop as per requirement until all modified data on the home is prefetched to the cache and both sites are synchronized. You can run a separate prefetch for download the data on-demand.

1. On a home cluster, which is running IBM Storage Scale, issue the following command:

```
mmafmctl fs1 getOutbandList -j home1 --path /gpfs/fs1/home1
```

A sample output is as follows:

```
Run prefetch, ex. --list-file /gpfs/fs1/home1/.mmmigrateCfg/.mmmigrate.changed.files
Run prefetch, ex. --delete --list-file /gpfs/fs1/home1/.mmmigrateCfg/.mmmigrate.deleted.files
```

The **getOutBandList** option generates list-files of all the files, directories on the home site, and their metadata exists on the specified path. This command generates two list-files, such as changed and deleted files.

- The deleted list-file contains information about the files/directories, which were deleted from the specified fileset path on the home fileset. This file can be used to delete the data from an AFM independent writer (IW), single writer (SW), local updates (LU), and read only (RO) mode fileset on the cache cluster.
- The changed list-file contains information about the files or directories and its metadata. The metadata can be used to pre-populate an AFM independent writer (IW), single writer (SW), local updates (LU), and read only (RO) mode fileset on the cache cluster by using the **prefetch --outband** option.

2. On the cache site by using the generated files issue the following command:

```
mmafmctl fs1 prefetch -j lu1 --list-file /gpfs/rfs1/
target1/.mmmigrateCfg/.mmmigrate.deleted.files --delete --home-fs-path target1
mmafmctl fs1 prefetch -j lu1 --outband --list-file /gpfs/rfs1/
target1/.mmmigrateCfg/.mmmigrate.changed.files --prefetch-threads 8
```



# Chapter 58. Administering AFM DR

The following topics assist you in Administering AFM DR.

## Enabling integrated archive manager (IAM) modes on AFM-DR filesets

To enable the Immutability and appendOnly options, you must enable an IAM mode on AFM-DR primary and secondary filesets.

1. Ensure that the AFM-DR primary fileset is in the Active state and no replication is in-progress.

```
mmafmctl fs1 getstate -j afmDR-Primary1
```

The sample output is as follows:

| Fileset Name   | Fileset Target                              | Cache State | Gateway Node | Queue Length | Queue numExec |
|----------------|---------------------------------------------|-------------|--------------|--------------|---------------|
| afmDR-Primary1 | nfs://192.168.1.1/gpfs/fs1/afmDR-Secondary1 | Active      | c80f5m5n05   | 0            | 1035          |

2. Stop the AFM-DR primary fileset.

```
mmafmctl fs1 stop -j afmDR-Primary1
```

3. Enable an IAM mode on a primary fileset.

```
mmafmctl fs1 afmDR-Primary1 --iam-mode ad
```

4. Enable an IAM mode on a secondary fileset.

```
mmafmctl fs1 afmDR-Secondary1 --iam-mode ad
```

5. Start the primary fileset.

```
mmafmctl fs1 start -j afmDR-Primary1
```

**Note:** On a secondary fileset, you need not to run the stop or start command.

The start or stop command must be run only on an AFM-DR primary fileset.

6. List the IAM mode that is set on the AFM-DR fileset.

```
mmfileset fs1 afmDR-Primary1 --iam-mode
```

The sample output is as follows:

| Filesets in file system 'fs1': |        |           |          |
|--------------------------------|--------|-----------|----------|
| Name                           | Status | Path      | IAM mode |
| afmDRPrimary1                  | Linked | /gpfs/fs1 | advisory |

**Note:** The same IAM mode must reflect on the secondary site.

## Creating an AFM-based DR relationship

Use this procedure to create and use an AFM-based Async DR relationship:

1. Create a primary fileset. Run on a primary cluster.

Create the primary fileset by using the **mmcrfileset** command. The primary can be connected to the secondary using NFSv3 protocol or the NSD protocol. All AFM parameters for writable filesets (single writer or independent writer) are applicable to a primary fileset. A primary fileset is not revalidated and

does not check the secondary for changes because it is expected that changes always originate from the primary.

A primary fileset is a writable fileset. Therefore, all file operations that are performed on this fileset are replayed at the secondary fileset by using the same mechanism as single writer and independent writer modes. Unlike other AFM modes the secondary, or target fileset is an AFM fileset that has a relationship with a primary. The secondary fileset is enforced as read-only. AFM parameters such as **afmAsyncDelay**, number of flush threads, and parallel write can be used on primary filesets.

When a primary fileset is created, a unique primary ID is generated. When you create a primary fileset, you need to specify the path to the secondary fileset though it might not exist at the time of primary creation. In the following example, the secondary is not created but the path is provided in **mmcrfileset** command.

```
mmcrfileset fs1 primary2 -p afmMode=primary --inode-space=new -p
afmTarget=nfs://c2m3n06/ibm/fs1/secondary2 -p afmRPO=720
Fileset primary2 created with id 19 root inode 7340035.
Primary Id (afmPrimaryID) 15997179941099568310-C0A8747F557F0086-19
```

**Note:** If you are using CES NFS at home, replace *c2m3n06* with *ces\_ip\_of\_secondary\_node*.

2. Create a secondary fileset. Run on a secondary cluster.

Get the primary ID of the GPFS fileset on the primary side (**afmPrimaryID**) before the actual conversion. Use **mmafmctl getPrimaryId** command on the GPFS fileset on the primary side.

```
mmafmctl fs1 getPrimaryId -j primary2
Primary Id (afmPrimaryID) 15997179941099568310-C0A8747F557F0086-19
```

Create a secondary fileset by using the **mmcrfileset** command.

```
mmcrfileset fs1 secondary2 -p afmMode=secondary -p
afmPrimaryID=15997179941099568310-C0A8747F557F0086-19
--inode-space new
```

3. Link the secondary fileset on the secondary cluster by using the **mmlinkfileset** command.

Run **mmlinkfileset fs1 secondary2 -J /ibm/fs1/secondary2**.

The primary does not check the secondary for changes. If you are using quotas, ensure that the same value is set for quotas on primary and secondary. On a primary fileset, eviction is disabled by default and filesets do not expire. If you are using NFS, ensure that the NFS export on the secondary site is accessible from the gateway nodes in the primary cluster. If you are using the NSD protocol, the secondary file system needs to be mounted on the gateway nodes at the primary cluster.

4. Restart NFS on secondary.

**Note:** If you are using CES NFS, you need not to restart NFS.

5. Link the primary fileset on the primary cluster.

Link the primary fileset by using **mmlinkfileset** command. Linking the fileset creates the first RPO snapshot on the primary called **psnap0**.

```
mmlinkfileset fs1 primary2 -J /ibm/fs1/primary2
```

After the primary and secondary are linked, the RPO snapshot (**psnap0**) gets queued from the primary fileset, which gets replayed on the secondary fileset. The Async DR filesets are now ready for use.

- Do not run **mmafmconfig** command on the secondary site. Run **mmafmctl gpfs0 getstate** on the primary to know the primary gateway node.
- Check **fsid** and **primary id** on the secondary, and ensure that any two secondary filesets do not have the same **fsid** or **primary id**.
- **psnap0** must be created at both sites for the filesets to synchronize. In cases like node shutdown, process failure; **psnap0** might not be created. Hence, filesets do not synchronize with the secondary. Unlinking and re-linking the filesets re-creates **psnap0**. The filesets then synchronize with the secondary.

- **aFmRPO** value can be set according to the workload on the Primary fileset. In case of high workload, Primary fileset might see RPO miss.

## Converting GPFS filesets to AFM DR

---

The IBM GPFS-independent filesets to primary or secondary filesets conversion can be done for the replication.

Complete the following steps to convert IBM GPFS-independent filesets to primary or secondary:

1. By using the trucking method, ensure that the secondary site has the same data as the primary site.

An existing IBM GPFS-independent fileset can be converted to primary or secondary. If the fileset on the primary site has data, the secondary site must be synchronized with the same data. This process is called trucking. The trucking must be inband.

### Inband trucking

Copying the data from the primary to the secondary while you are setting up the relationship. The inband trucking is limited by the network bandwidth between the primary and the secondary.

The conversion of a regular independent fileset to AFM primary with the **mmafmctl** command must be performed by specifying the **--check-metadata** option. This option verifies that the fileset does not contain objects with attributes that are not allowed in a primary fileset. These objects are as follows:

- Special files (such as devices)
- Dependent fileset
- Clones where a source belongs to a snapshot.

2. Before the actual conversion, get the primary ID of the GPFS fileset by using the **mmafmctl getprimaryid** command on the GPFS fileset.

3. Convert the fileset on the secondary site to a secondary and set the primary ID by using the **mmchfileset** or **mmafmctl** command with the **convertToSecondary** option.

If NFS is used to define an AFM target on the primary site, ensure that the NFS export on the secondary site is accessible on the primary. If GPFS protocol is used for the target, the secondary file system must be remote-mounted on the primary site.

**Note:** If you are establishing the secondary by using the out-of-band option, do the following steps:

- a. Complete the data copy.
  - b. Ensure that the primary and the secondary have the same data before you configure the secondary with the unique ID of the primary.
4. Restart the NFS server on the secondary site. If the NFS server is used by other application, reexport the directories on the secondary site by using the **exportfs -ra** option.
  5. Convert the fileset on the primary site to a primary by using **mmafmctl** command. Run on the primary cluster. Gateway nodes must be defined in the primary site and the file system must be mounted on all gateway nodes before this conversion. Run the **mmafmctl** command with the **convertToPrimary** option.

```
mmafmctlDevice convertToPrimary-j FilesetName
[--afmtarget Target { --inband | --secondary-snapshotname SnapshotName }
[--check-metadata | --nocheck-metadata] [--rpo RPO] [-s LocalWorkDirectory]
```

The **--afmtarget** option and the **--inband** option are mandatory options for the first conversion command on the fileset. The conversion can be interrupted due to unforeseen reasons or in case of rare errors when psnap0 creation is not successful. In such cases, the fileset is converted to a primary but the state is **PrimInitFail**.

Based on the reason behind the failure, the administrator must rerun the conversion command without any argument. Alternatively, the fileset can be converted back to normal GPFS filesets and converted again by using the conversion command with arguments.

The **--afmtarget** option mentions the fileset path on the secondary site.

The `--inband` option is used for the inband trucking. Primary ID is generated and the first RPO snapshot `psnap0` is created. The entire data on the snapshot is queued to the secondary. After the data is replayed on the secondary after Step 3 (following), that is, the primary and secondary are connected, it creates a `psnap0` snapshot on the secondary ensuring that the `psnap0` on the primary and the secondary are the same. Now, you can consider that a relationship is established.

Before the conversion, the `--check-metadata` option checks if the disallowed types such as immutable/append-only files, clones where the source belongs to a snapshot, are present in the GPFS fileset on the primary site. If the disallowed types still exist on the primary site, the conversion fails. The `--check-metadata` option is not mandatory and scans the entire fileset to verify its contents. If the fileset is known to be permissible for conversion, it can be excluded and it must be used whenever you have a doubt. This option is the default option.

The `--no_check-metadata` option is used to proceed with conversion without checking for the disallowed types.

The `--rpo` option specifies the RPO interval in minutes for this primary fileset. By default, RPO is disabled. You can use the `mmchfileset` command to modify the `afmRPO` parameter value of the AFM DR fileset. The `--secondary-snapname` option is not applicable for the AFM or GPFS filesets conversion. This option is used while you are establishing a new primary, as discussed in subsequent sections.

Gateway node assignments must be finalized and done preferably before conversion of GPFS or AFM filesets to the primary site. If no gateway is present on the primary cluster during conversion, then primary fileset might remain in the `PrimInitFail` state.

After the primary and secondary are connected with `psnap0` from any one side, the primary is in Active state. The two filesets are ready for use.

For more information, see `mmafmctl` command in *IBM Storage Scale: Command and Programming Reference Guide*.

**Note:** Parallel data transfers are not applicable to trucking even if the AFM target is mapping. Resync does not split data transfers even if parallel data transfer is configured, and the target is a mapping.

## Converting AFM relationship to AFM DR

A working AFM single writer (SW) or independent writer (IW) relationship can be converted to a primary or secondary relationship.

Complete the following steps:

**Note:** In case of multiple IW caches to the same home, you can convert only one to primary.

1. Ensure that all contents are cached. An AFM fileset must be in the active state by flushing queues and for filesets that have contents on home, the complete namespace must be constructed on the cache by using stat on all entries to avoid orphans. In SW/IW filesets, some files might not be cached or some files might be evicted. All such files must be cached by using prefetch. Ensure that all contents are present and are up to-date in the SW/IW caches.

- a) Ensure that the storage capacity on the cache fileset is the same as on the home and the set quotas match.
- b) Disable the automatic eviction.

```
mmchfileset filesystem sw/iw cache -p afmEnableAutoEviction=no
```

- c) Ensure that `afmPrefetchThreshold` is set to 0 on the SW/IW cache.
- d) Run a policy scan on the home to get the list of files, and use the list in `mmafmctl prefetch` on the cache to ensure that all files are cached. Or, run a policy scan on the cache to test the cached flag of each file and report on any that are not fully cached.

2. Convert the fileset on the primary site to a primary using `mmafmctl`. The primary ID is generated and a `psnap0` is created on the primary site. AFM gateway nodes must be defined in the primary site, and the file system is mounted on all gateway nodes before conversion. By default, RPO is

disabled. You can use the **mmafmctl convertToPrimary** command to enable RPO. You can use the **mmchfileset** command later to enable RPOs.

3. Convert the home to a secondary and set the primary ID by using **mmchfileset** or **mmafmctl** with the **convertToSecondary** option. Run on the primary cluster.

After the primary and secondary are converted and connected through primary ID, the psnap0 queued from the primary fileset is played on the secondary fileset. The two filesets are ready for use. For more information, see *mmafmctl* command in *IBM Storage Scale: Command and Programming Reference Guide*.

**Note:**

- IW/SW fileset must communicate at-least once to home before conversion. Newly created and inactive filesets might not convert successfully. When you convert a fileset in the inactive state, it will convert to primary but will not create psnap0. Next access of the primary fileset will trigger recovery and create the psnap0 and move the psnap0 and the pending changes to home.
- If applications are in progress on the cache fileset during conversion, some inodes might be orphans and the --check-metadata option might show failures. It might be useful to use the --nocheck-metadata option in such cases.
- If cached files had been evicted from SW/IW cache, conversion with the --check-metadata option might show failures. It might be useful to use the --nocheck-metadata option in such cases.
- If home of an IW fileset is running applications during conversion, IW must revalidate with home to pull in all the latest data before conversion. During conversion, if any file or directory is not in the cache, it might result in a conflict error and fileset might go into the NeedsResync state. AFM automatically fixes the conflicts during the next recovery.
- You cannot convert an SW fileset that is in an unmounted state or the NeedsResync state.
- Resync does not split data transfers even if parallel data transfer is configured, and the target is a mapping.



# Chapter 59. Administering AFM to cloud object storage

## Managing AFM to cloud object storage keys

Use the **mmafmcoskeys** command for the keys management in the AFM to cloud object storage. Each object has data, metadata, and keys. The object key (or key name) uniquely identifies the object in a bucket.

An access key and a secret key are needed to access a bucket on a cloud object server. By using the **mmafmcoskeys** command, the keys administration becomes simple. You can set these keys for specific buckets on specific servers. The keys can either be specified by using the command line or can be provided as an input key file. In this input key file, on each line an access key and a secret key are separated with a colon.

After the access key and storage key are set, AFM to cloud object storage reads the key when you set up the relationship and connect to the cloud object storage for the first time. In case if cloud object storage expires the keys, then AFM is unable to access the data from the server. You can also set the expiration timeout value that force AFM to refresh or reload the keys into the memory and use the key for communication with cloud object storage server. You must update the keys after expiration and before you start the next communication with server. To set the object key expiration timeout in seconds, issue the following command:

```
mmchconfig afmObjKeyExpiration=1800 -i
```

You can also get a report of all access keys or secret keys that are stored for a bucket by using the **mmafmcoskeys** command. This report has a list of all keys across the cluster.

The following example shows how you can manage access and secret keys by using the **mmafmcoskeys** command. For more information, see the *mmafmcoskeys command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

1. Obtain access and secret keys for a bucket from a cloud object provider, for example, Amazon S3, Microsoft Azure Blob, and IBM Cloud Object Storage.

In this example, the following keys are used:

AccessKey = key1234567890

SecretKey = key1234567890

2. Set the keys that are obtained from the cloud object provider by issuing the following command:

```
mmafmcoskeys newbucket:192.0.2.* set key1234567890 key1234567890
```

where,

**newbucket**

Specifies a bucket name.

**192.0.2.\***

IP of a server.

**Note:** If you do not set the keys for a bucket before the AFM to cloud object storage relation is set, the **mmafmcosconfig** command fails.

3. Establish the AFM to cloud object storage relation by issuing the following command:

```
mmafmcosconfig fs1 newbucket --endpoint http://192.0.2.* --uid 0 --gid 0 --new-bucket newbucket --mode sw --cleanup --object-fs
```

4. Display specific keys by issuing the following command:

```
mmafmcoskeys newbucket:192.0.2.* get
key1234567890:key1234567890
```

- Get the report of all keys that are stored on the AFM to cloud object storage by issuing the following command:

```
mmafmcoskeys all get --report
```

A sample output is as follows:

```
bucket2:lb1.ait.examplelabs.com=COS:BCGSt6BBCDqLowpVF2zd:lcxHFFYWB8XG1noeQDJP1GoHC2khBY8gr1RQ
05Cv
bucket3.1:lb1.ait.examplelabs.com=COS:BCGSt6BBCDqLowpVF2zd:lcxHFFYWB8XG1noeQDJP1GoHC2khBY8gr1
RQ05Cv
my.name=COS:key1234567890:key1234567890
newbucket:192.0.2.*=COS:key1234567890:key1234567890
```

- If a bucket is removed or updated by using the delete option, delete access and secret keys by issuing the following command:

```
mmafmcoskeys newbucket:192.0.2.* delete
```

## Creating AFM to cloud object storage relation in different modes

You can set an IBM Storage Scale cluster to create AFM to cloud object storage filesets to connect to cloud object services such as Amazon S3, Microsoft Azure Blob, and IBM Cloud Object Storage. To set an AFM to cloud object storage relation, you can use read-only (RO), single-writer (SW), local-update (LU), and independent-writer (IW) modes in the AFM to cloud object storage.

When you create an AFM to cloud object storage relation by using the **mmafmcosconfig** command, use the **--mode** option to define the mode of operation for the relation.

### Read-only (RO)

In this mode, data in an AFM to cloud object storage fileset is read-only. You cannot create or modify objects in the fileset. If an application uses an object while it is being re-created (deleted and re-created with the same name) on a cloud object storage server, it is re-created in the cache. When the RO-mode fileset is set, download or prefetch can bring the data of necessary objects, or objects can be brought from the cloud object storage to the fileset on-demand while objects are being read.

**Note:** When an RO-mode AFM to cloud object storage relation is set, any metadata operation such as **ls** populates the fileset with metadata.

Example:

- Create an RO-mode AFM to cloud object storage relation.

```
mmafmcosconfig fs1 readonly --endpoint http://c1f1u11n07
--uid 0 --gid 0 --new-bucket readonly --mode ro --cleanup --object-fs
```

- Determine the RO-mode of the fileset.

```
mm1sfileset fs1 readonly --afm -L
```

A sample output is as follows:

```
Filesets in file system 'fs1':
Attributes for fileset readonly:
=====
Status
Path
Id
Root inode
Parent Id
Created
Comment
Inode space
Maximum number of inodes
Linked
/gpfs/fs1/readonly
1
524291
0
Tue Oct 20 08:53:44 2020
1
100352
```

```

Allocated inodes 100352
Permission change flag chmodAndSetacl
afm-associated Yes
Target http://c1f1u11n07:80/readonly
Mode read-only
File Lookup Refresh Interval 120
File Open Refresh Interval 120
Dir Lookup Refresh Interval 120
Dir Open Refresh Interval 120
Async Delay disable
Expiration Timeout disable (default)
Last pSnapId 0
Display Home Snapshots yes (default)
Parallel Read Chunk Size 0
Number of Gateway Flush Threads 16
Prefetch Threshold 0 (default)
Eviction Enabled yes (default)
IO Flags 0x0 (default)

```

3. Create obj1, obj2, and obj3 objects on a cloud object storage.

4. Check whether the objects are cached.

```
ls -lsh /gpfs/fs1/readonly/
```

A sample output is as follows:

```

total 0
0 -rwx----- 1 root root 13 Oct 20 2020 obj1
0 -rwx----- 1 root root 13 Oct 20 2020 obj2
0 -rwx----- 1 root root 13 Oct 20 2020 obj3

```

5. Pull objects data from the cloud object storage to the AFM to cloud object storage fileset.

```
cat /gpfs/fs1/readonly/obj1
```

A sample output is as follows:

```
111111111111
```

```
cat /gpfs/fs1/readonly/obj2
```

A sample output is as follows:

```
111111111111
```

6. Check the fileset status.

```
ls -lsh /gpfs/fs1/readonly/
```

A sample output is as follows:

```

total 1.0K
512 -rwx----- 1 root root 13 Oct 20 2020 obj1
512 -rwx----- 1 root root 13 Oct 20 2020 obj2
0 -rwx----- 1 root root 13 Oct 20 2020 obj3

```

**Note:** Data of the obj3 object is not read. Therefore, the object is uncached in the fileset.

### Single-writer (SW)

In this mode, only an AFM to cloud object storage fileset can do all the write operation or data is generated on the SW-mode fileset and the fileset does not check the cloud object storage for file or object updates. Ensure that no write operation is done on the cloud object storage server.

**Note:** You cannot enforce this check.

A cloud object storage server can have some pre-existing data. The SW-mode fileset can replicate this data by using the download or prefetch operation, and this data can be updated. Any updates in the fileset data are queued on the gateway node and passed to the cloud object storage server. For

more information about the download operation, see *mmafmcosctl command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

Example:

1. Create an SW-mode AFM to cloud object storage relation.

```
mmafmcosconfig fs1 singlewriter --endpoint http://c1f1u11n07
--uid 0 --gid 0 --new-bucket singlewriter --mode sw --cleanup --object-fs
```

2. Check whether the objects are cached.

```
ls -lsh /gpfs/fs1/singlewriter/
```

A sample output is as follows:

```
total 0
```

3. Create the objects in an IBM Storage Scale cluster.

```
for a in `seq 3`; do dd if=/dev/urandom of=/gpfs/fs1/singlewriter/object$a count=12
bs=256K; done
```

A sample output is as follows:

```
12+0 records in
12+0 records out
3145728 bytes (3.1 MB, 3.0 MiB) copied, 0.0155518 s, 202 MB/s
12+0 records in
12+0 records out
3145728 bytes (3.1 MB, 3.0 MiB) copied, 0.0150043 s, 210 MB/s
12+0 records in
12+0 records out
3145728 bytes (3.1 MB, 3.0 MiB) copied, 0.0160235 s, 196 MB/s
```

4. Check the cache state.

```
mmafmcctl fs1 getstate
```

A sample output is as follows:

| Fileset      | Name | Fileset | Target                            | Cache | State  | Gateway | Node    | Queue | Length | Queue | numExec |
|--------------|------|---------|-----------------------------------|-------|--------|---------|---------|-------|--------|-------|---------|
| singlewriter |      |         | http://c1f1u11n07:80/singlewriter |       | Active |         | c7f2n05 | 0     |        | 7     |         |

5. Check whether the objects are synchronized to a cloud object storage server from a cloud object storage GUI.

```
Name : object1
Date : 2020-10-20 08:54:06 EDT
Size : 3.0 MiB
ETag : b7a173514d704481128f96bae96c4735
Type : file
Metadata :
 Content-Type: application/octet-stream

Name : object2
Date : 2020-10-20 08:54:07 EDT
Size : 3.0 MiB
ETag : d983316f3457644c45f3db63fb496060
Type : file
Metadata :
 Content-Type: application/octet-stream

Name : object3
Date : 2020-10-20 08:54:07 EDT
Size : 3.0 MiB
ETag : 862841fbcc08eff03878230a0b32e7c71
Type : file
Metadata :
 Content-Type: application/octet-stream
```

## Independent-writer (IW)

This mode allows multiple AFM to cloud object storage filesets to point to the same cloud object storage bucket. Multiple AFM to cloud object storage filesets can be on the same IBM Storage Scale cluster or on a different cluster. Also, they point to the cloud object storage server. There is no synchronous locking between clusters when objects are updated on the cloud object storage server. Each AFM to cloud object storage fileset reads from the cloud object storage server and makes updates to the cloud object storage server independently. Reads and updates are based on the revalidation intervals and the asynchronous delay.

This mode is used to access different objects from each IW-mode AFM to cloud object storage. For example, unique users on each site are updating objects in their cloud object storage server bucket. Although this mode allows multiple AFM to cloud object storage clusters to modify the same objects, only advanced users must modify the objects because there is no locking or ordering between updates. Updates are propagated to the cloud object storage server in an asynchronous manner and can be delayed because of network disconnections. Therefore, conflicting updates from multiple AFM to cloud object storage sites can cause the data on the cloud object storage server to be undetermined.

Example:

1. Create an IW-mode AFM to cloud object storage relation.

```
mmafmcosconfig fs1 indwriter --endpoint http://c1f1u11n07
--uid 0 --gid 0 --new-bucket indwriter --mode iw --cleanup --object-fs
```

2. Check whether objects are cached.

```
ls -lsh /gpfs/fs1/indwriter/
```

A sample output is as follows:

```
total 0
```

3. Create objects in an IBM Storage Scale cluster.

```
for a in `seq 3`; do dd if=/dev/urandom of=/gpfs/fs1/indwriter/object$a count=12
bs=256K; done
```

A sample output is as follows:

```
12+0 records in
12+0 records out
3145728 bytes (3.1 MB, 3.0 MiB) copied, 0.0158949 s, 198 MB/s
12+0 records in
12+0 records out
3145728 bytes (3.1 MB, 3.0 MiB) copied, 0.0150224 s, 209 MB/s
12+0 records in
12+0 records out
3145728 bytes (3.1 MB, 3.0 MiB) copied, 0.0153003 s, 206 MB/s
```

4. Check the fileset status.

```
ls -lsh /gpfs/fs1/indwriter/
```

A sample output is as follows:

```
total 9.0M
3.0M -rw-r--r-- 1 root root 3.0M Oct 20 09:04 object1
3.0M -rw-r--r-- 1 root root 3.0M Oct 20 09:04 object2
3.0M -rw-r--r-- 1 root root 3.0M Oct 20 09:04 object3
```

```
On COS:
Name : object1
Date : 2020-10-20 08:59:04 EDT
Size : 3.0 MiB
ETag : a1e25de2378c86479323de2345422923
Type : file
Metadata :
```

```

Content-Type: application/octet-stream

Name : object2
Date : 2020-10-20 08:59:05 EDT
Size : 3.0 MiB
ETag : 1bfa4345ba48dffdacc7037ce57cb112
Type : file
Metadata :
 Content-Type: application/octet-stream
Name : object3
Date : 2020-10-20 08:59:05 EDT
Size : 3.0 MiB
ETag : c8d7c7a1da270b2ab5d2675083842326
Type : file
Metadata :
 Content-Type: application/octet-stream

```

5. Create obj1, obj2, and obj3 objects.
6. Check whether the objects are cached.

```
ls -lsh /gpfs/fs1/indwriter/
```

A sample output is as follows:

```

total 0
0 -rwx----- 1 root root 13 Oct 20 2020 obj1
0 -rwx----- 1 root root 13 Oct 20 2020 obj2
0 -rwx----- 1 root root 13 Oct 20 2020 obj3

```

7. Pull objects data from the cloud object storage to the AFM to cloud object storage fileset.

```
cat /gpfs/fs1/indwriter/obj1
```

A sample output is as follows:

```
111111111111
```

```
cat /gpfs/fs1/indwriter/obj2
```

A sample output is as follows:

```
111111111111
```

8. Check whether the objects are cached.

```
ls -lsh /gpfs/fs1/indwriter/
```

A sample output as follows:

```

total 1.0K
512 -rwx----- 1 root root 13 Oct 20 2020 obj1
512 -rwx----- 1 root root 13 Oct 20 2020 obj2
 0 -rwx----- 1 root root 13 Oct 20 2020 obj3

```

## Local updates (LU)

The LU-mode behaves in a similar way as the RO mode. However, you can create and modify objects in the AFM to cloud object storage fileset. Updates in the fileset are considered local to the AFM to cloud object storage and are decoupled from the corresponding object on the cloud object storage server. Local updates are never pushed back to the cloud object storage server. When an object is modified, during the revalidation operation, the object is not compared to the version on the cloud object storage server to verify whether it is up to date. Changes of this object on the cloud object storage server do not have an impact on the replicated copy of the object and the object on the cloud object storage server.

Behaviors with local objects:

In AFM to cloud object storage, LU-mode objects have one of the following states:

## **Uncached**

Objects on the cloud object storage server are shown in the AFM to cloud object storage as uncached. For these objects, only metadata is copied into the AFM to cloud object storage. The object does not reside on the AFM to cloud object storage, but only on the cloud object storage server. Changes on the cloud object storage server are reflected in the AFM to cloud object storage.

## **Cached or replicated**

If an uncached object is read in the AFM to cloud object storage or pre-fetched, the state of the object changes to replicated or cached. In the replicated state, all changes to the object on the cloud object storage server are reflected in the AFM to cloud object storage. The object resides on the AFM to cloud object storage.

## **Local**

Object data or metadata that is modified on AFM to cloud object storage becomes local to the AFM to cloud object storage. The replicated objects relationship to the object in the cloud object storage server is broken. Changes on the cloud object storage server are not reflected in the AFM to cloud object storage and object changes are not copied to the cloud object storage server.

**Note:** Objects can be downloaded from a cloud object storage server and uploaded back without affecting the LU-mode semantics. When the object is downloaded, the object is synchronized to a cloud object storage object.

Example:

1. Create an LU-mode AFM to cloud object storage relation.

```
mmafmcosconfig fs1 localupdates --endpoint http://c1f1u11n07
--uid 0 --gid 0 --new-bucket localupdates --mode lu --cleanup --object-fs
```

2. Create three objects that have per-existing data on a cloud object storage.

3. Check whether the objects are cached.

```
ls -lsh /gpfs/fs1/localupdates
```

A sample output is as follows:

```
total 0
0 -rwx----- 1 root root 13 Oct 20 2020 obj1
0 -rwx----- 1 root root 13 Oct 20 2020 obj2
0 -rwx----- 1 root root 13 Oct 20 2020 obj3
```

4. Pull objects data from the cloud object storage to the AFM to cloud object storage fileset.

```
cat /gpfs/fs1/localupdates/obj1
```

A sample output is as follows:

```
111111111111
```

```
cat /gpfs/fs1/localupdates/obj2
```

A sample output is as follows:

```
111111111111
```

```
cat /gpfs/fs1/localupdates/obj3
```

A sample output is as follows:

```
111111111111
```

5. Check the fileset status.

```
ls -lsh /gpfs/fs1/localupdates
```

A sample output is as follows:

```
total 1.5K
512 -rwx----- 1 root root 13 Oct 20 2020 obj1
512 -rwx----- 1 root root 13 Oct 20 2020 obj2
512 -rwx----- 1 root root 13 Oct 20 2020 obj3
```

6. Check the modified the contents.

```
echo 2222222 >> /gpfs/fs1/localupdates/obj1
echo 2222222 >> /gpfs/fs1/localupdates/obj2
echo 2222222 >> /gpfs/fs1/localupdates/obj3
```

A sample output is as follows:

```
Name : obj1
Date : 2020-10-20 09:10:39 EDT
Size : 13 B
ETag : 87b8769b874865e65a4525bfe9e56ba8
Type : file
Metadata :
 Content-Type: application/octet-stream

Name : obj2
Date : 2020-10-20 09:10:44 EDT
Size : 13 B
ETag : 87b8769b874865e65a4525bfe9e56ba8
Type : file
Metadata :
 Content-Type: application/octet-stream

Name : obj3
Date : 2020-10-20 09:10:49 EDT
Size : 13 B
ETag : 87b8769b874865e65a4525bfe9e56ba8
Type : file
Metadata :
 Content-Type: application/octet-stream
```

7. To push the created objects to the cloud object storage, upload the objects.

```
mmafmcosctl fs1 localupdates /gpfs/fs1/localupdates upload --all
```

A sample output is as follows:

```
Queued Failed TotalData
 (approx in Bytes)
 3 0 63
Object Upload successfully queued at the gateway.

on COS :
Name : obj1
Date : 2020-10-20 09:19:08 EDT
Size : 21 B
ETag : a29344969f1524d72a050e910bb20ab0
Type : file
Metadata :
 Content-Type: application/octet-stream

Name : obj2
Date : 2020-10-20 09:19:08 EDT
Size : 21 B
ETag : a29344969f1524d72a050e910bb20ab0
Type : file
Metadata :
 Content-Type: application/octet-stream

Name : obj3
Date : 2020-10-20 09:19:08 EDT
Size : 21 B
ETag : a29344969f1524d72a050e910bb20ab0
Type : file
Metadata :
 Content-Type: application/octet-stream
```

**Note:** `http://c1f1u11n07:80` cloud object storage endpoint is used in the relation examples.

Along with these modes of operations, the AFM to cloud object storage has two more behavioral modes. These modes are Object-FS and ObjectOnly.

### Object-FS

In the Object-FS mode AFM to cloud object storage fileset is synchronized to the cloud object storage. RO, LU, and IW modes of filesets synchronize metadata to and from the cloud object storage server. It includes operations such as `readdir` and `lookups` for the synchronization. Objects are downloaded when they are read on-demand or an application that is running on fileset is working on it. The ObjectFS mode-enabled fileset behaves in a similar way as an AFM fileset. For SW and IW modes-enabled fileset, the AFM to cloud object storage uploads files as objects to the cloud object storage server. For AFM RO, LU, and IW mode-enabled fileset, the AFM to cloud object storage automatically synchronizes objects from the cloud object storage server to the fileset as files. Enable this parameter if the AFM to cloud object storage fileset needs to behave in a similar way as an AFM mode fileset.

### ObjectOnly

With the ObjectOnly mode, the fileset does not automatically synchronize with cloud object storage server. This behavior is the default behavior of operation. You need to manually download the data or metadata from the cloud object storage server into the AFM to cloud object storage filesets by using the `mmafmcosctl` command. You can download the data or metadata while data is being transferred from the AFM to cloud object storage fileset to the cloud object storage server without any manual intervention (SW, IW mode).

In ObjectOnly mode, the performance is higher than ObjectFS mode because all file system operations are not enabled. As users traverse the directory or object tree of a fileset, the object information from the cloud object storage server is checked and updated as needed on the fileset.

AFM to cloud object storage operations are serviced either synchronously or asynchronously. Reads and revalidations are synchronous operations. Update operations from the fileset are asynchronous to cloud object storage server based on the `afmAsyncDelay` interval. All update operations from the writable filesets such as IW or SW mode are on the primary gateway. Queues in the primary gateway are pushed to cloud object storage server asynchronously based on the `afmAsyncDelay` interval.

**Note:** With the ObjectFS mode, objects can be readily read on-demand from the cloud object storage server. Whereas the ObjectOnly mode download and upload operations can be used for priority data synchronization based on the mode of AFM to cloud object storage fileset.

## Evicting files or objects data

Complete the following steps to evict data from files or objects:

1. To create an AFM to cloud object storage relation, issue the following command:

```
mmafmcosconfig fs1 afmtocos1 --endpoint http://IP --uid 0 --gid 0 --new-bucket afmtocos1 --mode sw
--object-fs --cleanup
afmobjfs=fs1 fileset=afmtocos1
bucket=afmtocos1 newbucket=afmtocos1 objectfs=yes dir=
policy= tmpdir= tmpfile= cleanup=yes mode=sw
xattr=no ssl=no acls=no gcs=no vhb=
bucketName=afmtocos1 serverName=IP
Linkpath=/gpfs/fs1/afmtocos1 target=http://IP/afmtocos1
endpoint=--endpoint http://IP
XOPT=-p afmObjectSubdir=yes -p afmParallelWriteChunkSize=0 -p afmParallelReadChunkSize=0
rc=0 mmdsh -vN c7f2n05 /usr/lpp/mmfs/bin/mmafmtransfer -v -t -a key -s key -b afmtocos1 -e http://IP
bcreate
```

2. Create objects or files in the afmtocos1 fileset.

```
for a in `seq 10`; do dd if=/dev/urandom of=/gpfs/fs1/afmtocos1/object$a count=8 bs=256K &
done
```

- a. To get the contents of the fileset, issue the following command:

```
ls -lsh /gpfs/fs1/afmtocos1/
```

A sample output is as follows:

```
total 20M
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object1
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object10
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object2
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object3
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object4
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object5
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object6
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object7
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object8
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object9
```

**Note:** The space consumed is 20M, that is, each object is consuming 2 Mb.

- b. To check the disk usage, issue the following command:

```
du -sh /gpfs/fs1/afmtocos1/
```

A sample output is as follows:

```
20M /gpfs/fs1/afmtocos1/
```

3. Evict an evict file that contains a list of files for the data eviction.

- a. To get the list of files an evict file, issue the following command:

```
cat /root/evictfile
```

A sample output is as follows:

```
/gpfs/fs1/afmtocos1/object1
/gpfs/fs1/afmtocos1/object2
/gpfs/fs1/afmtocos1/object3
/gpfs/fs1/afmtocos1/object4
/gpfs/fs1/afmtocos1/object5
```

- b. To evict files or objects, issue the following command:

```
mmafmcosctl fs1 afmtocos1 /gpfs/fs1/afmtocos1/ evict --object-list /root/evictfile
```

4. To check evicted data blocks and the disk size, issue the following command:

```
ls -lsh /gpfs/fs1/afmtocos1/
```

A sample output is as follows:

```
total 10M
0 -rw-r--r-- 1 root root 2.0M Sep 10 2020 object1
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object10
0 -rw-r--r-- 1 root root 2.0M Sep 10 2020 object2
0 -rw-r--r-- 1 root root 2.0M Sep 10 2020 object3
0 -rw-r--r-- 1 root root 2.0M Sep 10 2020 object4
0 -rw-r--r-- 1 root root 2.0M Sep 10 2020 object5
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object6
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object7
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object8
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object9
```

The **ls** command shows that evicted data blocks and the disk size is 0 for object1 through object5.

5. To check the disk usage, issue the following command:

```
du -sh /gpfs/fs1/afmtocos1/
```

A sample output is as follows:

```
10M /gpfs/fs1/afmtocos1/
```

Now the disk space is 10M because other 10M is evicted from five objects.

6. Bring back objects and files from a cloud object storage.

- a. Issue to the following command:

```
cd /gpfs/fs1/afmtocos1/
```

- b. To get the contents of the fileset, issue the following command:

```
cat object1 object2 object3 object4 object5 > /dev/null
```

- c. To check the cache state, issue the following command:

```
mmamfctl fs1 getstate
```

A sample output is as follows:

| Fileset Name | Fileset Target         | Cache State | Gateway Node | Queue Length | Queue numExec |
|--------------|------------------------|-------------|--------------|--------------|---------------|
| afmtocos1    | http://IP:80/afmtocos1 | Active      | c7f2n05      | 0            | 36            |

- d. To get the contents of the fileset, issue the following command:

```
ls -lsh /gpfs/fs1/afmtocos1/
```

A sample output is as follows:

```
total 20M
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object1
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object10
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object2
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object3
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object4
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object5
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object6
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object7
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object8
2.0M -rw-r--r-- 1 root root 2.0M Sep 10 2020 object9
```

- e. To check the disk usage, issue the following command:

```
du -sh /gpfs/fs1/afmtocos1/
```

A sample output is as follows:

```
20M /gpfs/fs1/afmtocos1/
```

Now the used disk space is again 20Mb. This output shows that files are read and data is brought back from the cloud object storage.

## Evicting files or objects metadata

Complete the following steps to evict metadata from files or objects:

1. To check a fileset, issue the following command:

```
mmlsfileset fs1 afmtocos1 -i
```

A sample output is as follows:

```
Filesets in file system 'fs1':
Name Status Path InodeSpace MaxInodes AllocInodes UsedInodes
afmtocos1 Linked /gpfs/fs1/afmtocos1 1 100352 100352 13
```

**Note:** The used inodes are 13. You need to evict 5 objects metadata.

2. To check the list of files in an evict file, issue the following command:

```
cat /root/evictfile
```

A sample output is as follows:

```
/gpfs/fs1/afmtocos1/osobject1
/gpfs/fs1/afmtocos1/osobject2
/gpfs/fs1/afmtocos1/osobject3
/gpfs/fs1/afmtocos1/osobject4
/gpfs/fs1/afmtocos1/osobject5
```

3. To evict the metadata, issue the following command:

```
mmafmcosctl fs1 afmtocos1 /gpfs/fs1/afmtocos1/ evict --object-list /root/evictfile --
metadata
```

4. To check the fileset again, issue the following command:

```
mm1sfileset fs1 afmtocos1-i
```

A sample output is as follows:

```
Filesets in file system 'fs1':
Name Status Path InodeSpace MaxInodes AllocInodes UsedInodes
afmtocos1 Linked /gpfs/fs1/afmtocos1 1 100352 100352 8
```

Metadata is evicted and the inodes are reduced to 8.

5. Populate the metadata after any metadata read operation is performed.

- a. To get the contents of the fileset, issue the following command:

```
ls -l /gpfs/fs1/afmtocos1/
```

A sample output is as follows:

```
total 10240
-rwx----- 1 root root 2097152 Sep 10 2020 osobject1
-rw-r--r-- 1 root root 2097152 Sep 10 2020 osobject10
-rwx----- 1 root root 2097152 Sep 10 2020 osobject2
-rwx----- 1 root root 2097152 Sep 10 2020 osobject3
-rwx----- 1 root root 2097152 Sep 10 2020 osobject4
-rwx----- 1 root root 2097152 Sep 10 2020 osobject5
-rw-r--r-- 1 root root 2097152 Sep 10 2020 osobject6
-rw-r--r-- 1 root root 2097152 Sep 10 2020 osobject7
-rw-r--r-- 1 root root 2097152 Sep 10 2020 osobject8
-rw-r--r-- 1 root root 2097152 Sep 10 2020 osobject9
```

- b. To check the fileset usage, issue the following command:

```
mm1sfileset fs1 afmtocos1 -i
```

A sample output is as follows:

```
Filesets in file system 'fs1':
Name Status Path InodeSpace MaxInodes AllocInodes UsedInodes
afmtocos1 Linked /gpfs/fs1/afmtocos1 1 100352 100352 13
```

Now the used inodes are 13 because all evicted metadata is brought back to the cloud object storage.

## Evicting data or objects by using the manual updates mode of the AFM to cloud object storage

The replication of the AFM to cloud object storage file system by using manual updates mode does not support automatic eviction, instead data or objects can be evicted by using **mmafmcosctl evict** command.

1. List the file systems.

```
ls -lash /gpfs/fs3
total 81M
256K drwxr-xr-x 4 root root 256K May 19 10:44 .
```

```

4.0K drwxrwxrwx 6 root root 4.0K May 19 03:29 ..
512 drwx----- 2 root root 4.0K May 19 03:35 .afm
8.0M -rwxr-xr-x 1 root root 8.0M May 19 10:44 file1
8.0M -rwxr-xr-x 1 root root 8.0M May 19 10:44 file10
8.0M -rwxr-xr-x 1 root root 8.0M May 19 10:44 file2
8.0M -rwxr-xr-x 1 root root 8.0M May 19 10:44 file3
8.0M -rwxr-xr-x 1 root root 8.0M May 19 10:44 file4
8.0M -rwxr-xr-x 1 root root 8.0M May 19 10:44 file5
8.0M -rwxr-xr-x 1 root root 8.0M May 19 10:44 file6
8.0M -rwxr-xr-x 1 root root 8.0M May 19 10:44 file7
8.0M -rwxr-xr-x 1 root root 8.0M May 19 10:44 file8
8.0M -rwxr-xr-x 1 root root 8.0M May 19 10:44 file9
512 drwx----- 65535 root root 4.0K May 19 03:35 .ptrash
512 dr-xr-xr-x 2 root root 8.0K Dec 31 1969 .snapshots

```

- Upload all files to the bucket on the cloud object storage.

```

mmafmcosctl fs3 root /gpfs/fs3/ upload --all
 Queued Failed TotalData
 (approx in Bytes)
 10 0 83588400
Object Upload successfully queued at the gateway.

```

- Create an evict file to evict these files.

```

cat > /evictfile
/gpfs/fs3/file1
/gpfs/fs3/file2
/gpfs/fs3/file3
/gpfs/fs3/file4
/gpfs/fs3/file5

```

- Check how many files are evicted.

```

cat > /evictfile
/gpfs/fs3/file1
/gpfs/fs3/file2
/gpfs/fs3/file3
/gpfs/fs3/file4
/gpfs/fs3/file5

```

- Check contents of the file system.

```

ls -lash /gpfs/fs3
total 41M
256K drwxr-xr-x 4 root root 256K May 19 10:44 .
4.0K drwxrwxrwx 6 root root 4.0K May 19 03:29 ..
0 -rwxr-xr-x 1 root root 8.0M May 19 10:44 file1
8.0M -rwxr-xr-x 1 root root 8.0M May 19 10:44 file10
0 -rwxr-xr-x 1 root root 8.0M May 19 10:44 file2
0 -rwxr-xr-x 1 root root 8.0M May 19 10:44 file3
0 -rwxr-xr-x 1 root root 8.0M May 19 10:44 file4
0 -rwxr-xr-x 1 root root 8.0M May 19 10:44 file5
8.0M -rwxr-xr-x 1 root root 8.0M May 19 10:44 file6
8.0M -rwxr-xr-x 1 root root 8.0M May 19 10:44 file7
8.0M -rwxr-xr-x 1 root root 8.0M May 19 10:44 file8
8.0M -rwxr-xr-x 1 root root 8.0M May 19 10:44 file9

```

Here, the file1 through file5 files are evicted.

## Mapping a directory to a cloud object storage bucket

You can map a directory to a bucket on a cloud object storage. This directory can be created under the fileset junction path.

You can retrieve the mapping information by using the get operation and delete the relation by using the delete operation.

- Create a relationship with a cloud object storage bucket by issuing the following command:

```

mmafmcosconfig fs1 singlewriter --endpoint http://c1f1u11n07 --uid 0 --gid 0
--bucket singlewriter --mode sw --object-fs

```

- Check the cache state by issuing the following command:

```
mmafmctl fs1 getstate
```

A sample output is as follows:

| Fileset Name | Fileset Target                    | Cache State | Gateway | Node    | Queue Length | Queue | numExec |
|--------------|-----------------------------------|-------------|---------|---------|--------------|-------|---------|
| singlewriter | http://c1f1u11n07:80/singlewriter | Active      |         | c7f2n05 | 0            |       | 5       |

- Map a directory to a cloud object storage bucket by using the access keys and secret keys of the cloud object storage bucket.

```
mmafmcosaccess fs1 singlewriter /gpfs/fs1/singlewriter/dir1 set --bucket cosbucket
--endpoint http://c1f1u11n07 --akey mkey1234 --skey mkey1234
```

**Note:** The `--keyfile` option can be used to specify a key file that contains an access key and a secret key. Instead of providing the access key and the secret key on a command line, you can use a key file. The key file must contain two lines for akey and skey separated by a colon.

- Check the cache state by issuing the following command:

```
mmafmctl fs1 getstate
```

A sample output is as follows:

| Fileset Name | Fileset Target                    | Cache State | Gateway | Node    | Queue Length | Queue | numExec |
|--------------|-----------------------------------|-------------|---------|---------|--------------|-------|---------|
| singlewriter | http://c1f1u11n07:80/singlewriter | Active      |         | c7f2n05 | 0            |       | 5       |

- Create objects by issuing the following commands:

```
touch /gpfs/fs1/singlewriter/dir1/object1
touch /gpfs/fs1/singlewriter/dir1/object2
touch /gpfs/fs1/singlewriter/dir1/object3
```

- Check the cache state by issuing the following command:

```
mmafmctl fs1 getstate
```

A sample output is as follows:

| Fileset Name | Fileset Target                    | Cache State | Gateway | Node    | Queue Length | Queue | numExec |
|--------------|-----------------------------------|-------------|---------|---------|--------------|-------|---------|
| singlewriter | http://c1f1u11n07:80/singlewriter | Dirty       |         | c7f2n05 | 3            |       | 8       |

- Check that these objects are played to a cloud object storage bucket.

```
Name : object1
Date : 2020-10-23 09:44:37 EDT
Size : 0 B
ETag : d41d8cd98f00b204e9800998ecf8427e
Type : file
Metadata :
 Content-Type: application/octet-stream
```

## Uploading objects

You can upload objects that are in the LU-mode.

Complete the following steps to upload an object:

- Create LU-mode objects on a cloud object storage.

```
mmafmcosconfig fs1 localupdates --endpoint http://c1f1u11n07
--uid 0 --gid 0 --new-bucket localupdates --mode lu --cleanup
```

A sample output is as follows:

```
Created 3 objects on COS:
Name : obj1
Date : 2020-10-23 06:07:29 EDT
Size : 2.2 KiB
ETag : 6fdfdb7af5eb5ad7f610014985951941
Type : file
Metadata :
 Content-Type: application/x-sh

Name : obj2
Date : 2020-10-23 06:07:31 EDT
Size : 2.2 KiB
ETag : 6fdfdb7af5eb5ad7f610014985951941
Type : file
Metadata :
 Content-Type: application/x-sh

Name : obj3
Date : 2020-10-23 06:07:33 EDT
Size : 2.2 KiB
ETag : 6fdfdb7af5eb5ad7f610014985951941
Type : file
Metadata :
 Content-Type: application/x-sh
```

2. Download the created objects.

```
mmafmcosctl fs1 localupdates /gpfs/fs1/localupdates/ download --all
```

A sample output is as follows:

```
Queued Failed AlreadyCached TotalData
 (approx in Bytes)
 3 0 0 6816
Object Download successfully queued at the gateway.
```

3. Check the fileset status.

```
mmafmctl fs1 getstate
```

A sample output is as follows:

| Fileset      | Name | Fileset Target                    | Cache State | Gateway Node | Queue Length | Queue numExec |
|--------------|------|-----------------------------------|-------------|--------------|--------------|---------------|
| localupdates |      | http://c1f1u11n07:80/localupdates | Active      | c7f2n05      | 0            | 10            |

4. Write data into objects, which will not be synchronized to a cloud object storage.

```
echo 11111 >> /gpfs/fs1/localupdates/obj1
echo 11111 >> /gpfs/fs1/localupdates/obj2
```

5. Check the fileset status.

```
mmafmctl fs1 getstate
```

A sample output is as follows:

| Fileset      | Name | Fileset Target                    | Cache State | Gateway Node | Queue Length | Queue numExec |
|--------------|------|-----------------------------------|-------------|--------------|--------------|---------------|
| localupdates |      | http://c1f1u11n07:80/localupdates | Active      | c7f2n05      | 0            | 10            |

These changes are local and are not pushed to the cloud object storage because it is in the LU-mode.

6. To synchronize objects to a cloud object storage, upload the objects.

```
mmafmcosctl fs1 localupdates /gpfs/fs1/localupdates upload --all
```

Sample outputs are as follows:

```
Queued Failed TotalData
 (approx in Bytes)
```

```
3 0 6828
Object Upload successfully queued at the gateway.
```

```
Objects are synced to COS:
Name : obj1
Date : 2020-10-23 06:15:01 EDT
Size : 2.2 KiB
ETag : 5a210361a71c03cb29565c4c2ae613cb
Type : file
Metadata :
 Content-Type: application/octet-stream

Name : obj2
Date : 2020-10-23 06:15:01 EDT
Size : 2.2 KiB
ETag : 5a210361a71c03cb29565c4c2ae613cb
Type : file
Metadata :
 Content-Type: application/octet-stream
```

## Downloading objects

Download of an object is important when the AFM to cloud object storage relation is created in the ObjectOnly mode. With the ObjectOnly mode, the fileset does not automatically synchronize with a cloud object storage. This behavior is the default behavior of operation.

You need to manually download data or metadata from the object storage server into the AFM to cloud object storage filesets by using the **mmafmcosctl** command. The data transfer from a fileset to an object storage server does not need any manual intervention (single-writer (SW), independent-writer (IW) mode).

When selected objects are needed for an application to run on an AFM to cloud object storage fileset, they can be downloaded by using the --objectlist option. All objects can be download by using the --all option. These objects are prefetched or downloaded from a cloud object storage.

The modifications to objects on a cloud object storage are not synchronized to the cache. Therefore, you need to download the objects. The modification to objects in IW and SW-modes is pushed to the cloud object storage.

When you download an object from AFM to cloud object storage or list objects, many objects or nested directories on a cloud object storage have extended attributes. The read directory operation takes time to populate metadata because the extended attributes need to be fetched.

In many AFM to cloud object storage use cases such as running analytics, when extended attributes (xattrs) synchronization is not required at the fileset level, set the **afmObjectFastReaddir** parameter value to yes for the improved read directory performance. To set this parameter, you must stop a fileset and start the fileset by using **mmafmctl** and **mmchfileset** commands.

When the **afmObjectFastReaddir** parameter value is set to yes, the AFM to cloud object storage read directory operation ingests metadata faster. Therefore, the metadata and later data read performance is improved. Extended attributes are not fetched from a cloud object storage. Also, objects that are deleted on a cloud object storage are not reflected in an AFM cache.

### Failed Object list

While downloading objects, due to network errors some objects can fail to download. To resolve this problem, system administrators can use --enable-failed-file-list option from **mmafmcosctl** download command.

Enabling --enable-failed-file-list option creates a list of failed objects on the gateway node or node specified in the command. System administrators can run the download command again by using **--object-list** parameter and the failed file list is generated to download the objects again.

1. Create an ObjectOnly IW-mode fileset.

```
mmafmcosconfig fs1 indwriter --endpoint http://c1f1u11n07
--uid 0 --gid 0 --new-bucket indwriter --mode iw
```

2. Check the fileset status.

```
mmlsfileset fs1 indwriter --afm -L
```

A sample output is as follows:

```
Filesets in file system 'fs1':

Attributes for fileset indwriter:
=====
Status Linked
Path /gpfs/fs1/indwriter
Id 1
Root inode 524291
Parent Id 0
Created Fri Oct 23 05:17:32 2020
Comment
Inode space 1
Maximum number of inodes 100352
Allocated inodes 100352
Permission change flag chmodAndSetacl
afm-associated Yes
Target http://c1f1u11n07:80/indwriter
Mode independent-writer
File Lookup Refresh Interval disable
File Open Refresh Interval disable
Dir Lookup Refresh Interval disable
Dir Open Refresh Interval disable
Async Delay 15 (default)
Last pSnapId 0
Display Home Snapshots no
Parallel Read Chunk Size 0
Number of Gateway Flush Threads 16
Prefetch Threshold 0 (default)
Eviction Enabled yes (default)
Parallel Write Chunk Size 0
IO Flags 0x400000 (afmSkipHomeRefresh)
```

3. Check whether you can write a file from the fileset.

```
touch /gpfs/fs1/indwriter/dd
```

4. Check whether the data is synchronized.

```
mmfmctl fs1 getstate
```

A sample output is as follows:

| Fileset Name | Fileset Target                 | Cache State | Gateway Node | Queue Length | Queue numExec |
|--------------|--------------------------------|-------------|--------------|--------------|---------------|
| indwriter    | http://c1f1u11n07:80/indwriter | Dirty       | c7f2n05      | 1            | 0             |

5. Check the fileset status.

```
mmfmctl fs1 getstate
```

A sample output is as follows:

| Fileset Name | Fileset Target                 | Cache State | Gateway Node | Queue Length | Queue numExec |
|--------------|--------------------------------|-------------|--------------|--------------|---------------|
| indwriter    | http://c1f1u11n07:80/indwriter | Active      | c7f2n05      | 0            | 1             |

The created object is synchronized.

6. Create objects on a cloud object storage.

```
Name : dd
Date : 2020-10-23 05:13:59 EDT
Size : 0 B
ETag : d41d8cd98f00b204e9800998ecf8427e
Type : file
Metadata :
 Content-Type: application/octet-stream

Name : obj1
```

```

Date : 2020-10-23 05:20:48 EDT
Size : 2.2 KiB
ETag : 6fdfdb7af5eb5ad7f610014985951941
Type : file
Metadata :
 Content-Type: application/x-sh

Name : obj2
Date : 2020-10-23 05:20:50 EDT
Size : 2.2 KiB
ETag : 6fdfdb7af5eb5ad7f610014985951941
Type : file
Metadata :
 Content-Type: application/x-sh

Name : obj3
Date : 2020-10-23 05:20:51 EDT
Size : 2.2 KiB
ETag : 6fdfdb7af5eb5ad7f610014985951941
Type : file
Metadata :
 Content-Type: application/x-sh

```

7. Generate a list file of the created objects.

```
cat /listfile
/gpfs/fs1/indwriter/obj1
/gpfs/fs1/indwriter/obj2
```

8. Download the objects.

```
mmafmcosctl fs1 indwriter /gpfs/fs1/indwriter download --object-list /listfile
```

A sample output is as follows:

| Queued | (Total) | Failed | AlreadyCached | TotalData<br>(approx in Bytes) |
|--------|---------|--------|---------------|--------------------------------|
| 2      | (3)     | 0      | 0             | 4544                           |

Object Download successfully queued at the gateway.

9. Ensure that data is downloaded.

```
ls -lash /gpfs/fs1/indwriter/
```

A sample output is as follows:

```

total 260K
 512 drwx----- 5 root root 4.0K Oct 23 05:54 .
256K drwxr-xr-x 4 root root 256K Oct 23 05:18 ..
 512 drwx----- 65535 root root 4.0K Oct 23 05:17 .afm
 512 drwx----- 65535 root root 4.0K Oct 23 05:17 .pconflicts
 512 drwx----- 65535 root root 4.0K Oct 23 05:17 .ptrash
 512 dr-xr-xr-x 2 root root 8.0K Dec 31 1969 .snapshots
 0 -rw-r--r-- 1 root root 0 Oct 23 05:19 dd
 512 -rwx----- 1 root root 2.3K Oct 23 2020 obj1
 512 -rwx----- 1 root root 2.3K Oct 23 2020 obj2

```

The objects are downloaded.

## Downloading objects by using the outband method

The outband download method is an enhanced method for downloading files or objects into AFM to cloud object storage filesets.

The outband download method uses multiple threads and direct **mmafmtransfer** interface, which negates the effect of serializing operations by VFS layer for high latency networks and increases the overall performance for download. In this method any node that can connect to cloud object storage can be used to download the files.

Multiple download instances can be run on multiple filesets by using multiple nodes that can connect to cloud object storage buckets.

When selected objects are needed for an application to run on an AFM to cloud object storage fileset, they can be downloaded by using the `--object-list` option. All objects can be download by using the `--all` option. These objects are prefetched or downloaded from a cloud object storage.

The outband download command supports failed object list that means that when some objects are failed to download due to network error or other issue, the outband download command generates a list of those objects. An administrator can use this object list and download the objects again. Use `--enable-failed-object-list` options to enable this feature.

**Note:**

- When outband download method is used to download objects, an administrator can see the download progress, but queue is not built on gateway nodes and fileset can remain inactive. As gateway node queue is not built for outband download, the download stats that use command `mmafmcosctl` are not supported.
- When `-N "nongateway node"` is specified for download, though download progresses, the command does not display continuous stats. On gateway node, command progression is displayed.
- Outband download does not support mapped directory download and parallel IO download. Read sizes from applications are ignored. Default part size is 16 M with eight number of parts are downloaded in parallel.
- `--directory-object` option must be enabled while creating AFM to Cloud Object Storage relations for outband download.
- The object updated on the Cloud Object Storage bucket after the download can be downloaded again by using the outband download method.

**Example**

Bucket `ibmc1` hosted on the IBM cloud already have objects:

1. To create a new fileset, issue the following command.

```
mmafmcosconfig fs1 ibmc2 --endpoint https://s3.us-east.cloud-object-storage.appdomain.cloud --object-fs --bucket ibmc1 --mode ro --directory-object --fast-readdir
```

2. To download the object by using outband method, issue the following command:

```
mmafmcosctl fs1 ibmc2 /gpfs/fs1/ibmc2 download --all --outband --threads 8 --enable-failed-object-list
```

Output:

| Queued | (Total) | Failed | AlreadyCached | TotalData (Bytes) |
|--------|---------|--------|---------------|-------------------|
| 7800   | 11001   | 0      | 0             | 70890000          |
| 11001  | 11001   | 0      | 0             | 100000000         |

Download command completed

## Converting IBM Storage Scale independent fileset to manual update mode

You can convert the existing IBM Storage Scale independent fileset to manual update (MU) mode.

Before you convert the IBM Storage Scale independent fileset to MU mode, ensure that the type of data such as file names, file sizes, and file types must match to the data syntax and semantics that are supported by cloud object storage (COS) provider.

To convert the IBM Storage Scale independent fileset to MU mode, the existing independent fileset with available data can be used. The `mmafmcosconfig` with `--convert` option is used to convert the fileset to MU mode.

After the fileset is converted to MU mode, all or specific data can be uploaded to the COS bucket.

**Note:**

- With **mmafmcosconfig --convert** option, other options: **--dir**, **--uid**, **--gid**, and **--perm** are not supported.
- While converting or creating MU mode fileset, **mmafmcosconfig -cleanup** options must not be used.

The following example contains the steps that are used to convert IBM Storage Scale independent fileset to MU mode:

1. Create IBM Storage Scale independent fileset and link that fileset.

```
Node1 03:10:30 1] mmcrfileset fs1 specfileset --inode-space=new
Fileset specfileset created with id 11 root inode 3145731.
Node1 03:10:40 1]
Node1 03:10:47 1] mmlinkfileset fs1 specfileset -J /gpfs/fs1/specfileset
Fileset specfileset linked at /gpfs/fs1/specfileset
```

2. Create few data in the fileset.

```
Node1 03:10:48 1] echo "file data" > /gpfs/fs1/specfileset/file1
Node1 03:10:55 1] echo "file data" > /gpfs/fs1/specfileset/file2
Node1 03:10:57 1] echo "file data" > /gpfs/fs1/specfileset/file3
```

3. Set up keys from Cloud object storage for a new bucket called “bucketoncos”.

```
Node1 03:11:00 1] mmafmcoskeys bucketoncos:s3.us-east.cloud-object-storage.appdomain.cloud
set 779c2178d53a4497de8d6cdc72 ef08ac5bcc1b8c35a293a7fe24c2effe54d2beba0
```

4. Create MU fileset by using independent fileset “specfileset”.

```
Node1 03:12:26 1] mmafmcosconfig fs1 specfileset --endpoint http://s3.us-east.cloud-object-
storage.appdomain.cloud --new-bucket bucketoncos --object-fs --mode mu --xattr --convert
mmafmcosconfig: Fileset specfileset is not an AFM fileset.
Converting GPFS fileset to AFM manual-updates fileset...
Fileset specfileset changed.
Node1 03:12:38 1]
Node1 03:12:39 1]
```

5. Verify that converted fileset uses **mmlsfileset** command.

```
Node1 03:12:39 1] mmlsfileset fs1 specfileset --afm -L
Filesets in file system 'fs1':

Attributes for fileset specfileset:
=====
Status Linked
Path /gpfs/fs1/specfileset
Id 11
Root inode 3145731
Parent Id 0
Created Tue Jan 11 03:10:36 2022
Comment
Inode space 6
Maximum number of inodes 100352
Allocated inodes 100352
Permission change flag chmodAndSetacl
afm-associated Yes
Permission inherit flag inheritAclOnly
Target http://s3.us-east.cloud-object-
storage.appdomain.cloud:80/bucketoncos
Mode manual-updates
File Lookup Refresh Interval 120
File Open Refresh Interval 120
Dir Lookup Refresh Interval 120
Dir Open Refresh Interval 120
Async Delay disable
Last pSnapId 0
Display Home Snapshots no
Parallel Read Chunk Size 0
Number of Gateway Flush Threads 8
```

```

Prefetch Threshold 0 (default)
Eviction Enabled yes (default)
IO Flags 0x10080000 (afmObjectXattr,afmMUPromoted)
IO Flags2 0x0 (default)

```

6. Files available on the fileset.

```

Node1 03:12:58 1] ls -sh /gpfs/fs1/specfileset
total 1.5K
512 file1 512 file2 512 file3
Node1 03:13:16 1]

```

7. Create an objectlist file and upload the objects.

```

Node1 03:48:31 1] cat /root/objectlist
/gpfs/fs1/specfileset/file1
/gpfs/fs1/specfileset/file2

Node1 03:48:41 1] mmafmcosctl fs1 specfileset /gpfs/fs1/specfileset/ upload --object-list /
root/objectlist
 Queued (Total) Failed TotalData
 (approx in Bytes)
 2 (2) 0 20
Object Upload successfully queued at the gateway.
Node1 03:48:50 1]

```

8. Check the files stat by using -upload-stats.

```

Node1 03:49:12 1] mmafmcosctl fs1 specfileset --upload-stats
mmafmcosctl: Statistics of last or currently running upload are as follows:
Fileset Name Pending Failed Total Total data(Bytes)
Throughput(KB/s)
----- ----- ----- -----
----- -----
specfileset 0 0 2 20
0
Node1 03:49:27 1]

```

## Uploading and downloading files from MU mode fileset

The files can be uploaded and downloaded from cloud object storage by using AFM Manual Update (MU) mode.

MU mode fileset depends on **mmafmcosctl** upload and download commands to upload and download data to and from cloud object storage respectively.

Administrators can determine the files to upload and download from cloud object storage (COS). Files on AFM to cloud object storage MU fileset can be determined manually or by using policies and can be uploaded to the cloud object storage. Whereas, MU mode is not capable of automatically recognizing the new files that are added on COS by third-party interface or API. Therefore, administrators need to determine these files to download them into the MU fileset. To create and manage policies, see “[Policies for automating file management](#)” on page 531.

For upload and download, **mmafmcosctl** upload/download –object-list command needs to be used. The –object list can be created by using the absolute path of the objects with respect to the MU fileset.

**Note:** Use **mmafmcosctl** upload –all option of command with caution as this command with --all option uploads all the data to cloud object storage, and can overwrite the files on COS.

### Example 1: Upload

1. Create a new MU mode relation with COS bucket:

a. Set keys from cloud for the new bucket.

```

Node1] mmafmcoskeys demobucket-mu:s3.us-east.cloud-object-storage.appdomain.cloud
set 779c2178d4key497de8d6cdc72 ef08ac5bcc1c79f4key5a293a7fe24c2ef
fe54d2beba0
Node1]

```

b. Create MU fileset.

```
Node1] mmafmcosconfig fs1 demo_mu --endpoint http://s3.us-east.cloud-object-storage.appdomain.cloud --new-bucket demobucket-mu --object-fs --mode mu
```

2. Create files in MU mode fileset by using dd.

```
Node1] dd if=/dev/urandom of=/gpfs/fs1/demo_mu/file1 bs=256K count=4
4+0 records in
4+0 records out
1048576 bytes (1.0 MB, 1.0 MiB) copied, 0.0110485 s, 94.9 MB/s
Node1] dd if=/dev/urandom of=/gpfs/fs1/demo_mu/file2 bs=256K count=8
8+0 records in
8+0 records out
2097152 bytes (2.1 MB, 2.0 MiB) copied, 0.0124402 s, 169 MB/s
Node1] dd if=/dev/urandom of=/gpfs/fs1/demo_mu/file3 bs=256K count=12
12+0 records in
12+0 records out
3145728 bytes (3.1 MB, 3.0 MiB) copied, 0.0153719 s, 205 MB/s
Node1]
Node1] dd if=/dev/urandom of=/gpfs/fs1/demo_mu/file4 bs=256K count=16
16+0 records in
16+0 records out
4194304 bytes (4.2 MB, 4.0 MiB) copied, 0.0223167 s, 188 MB/s
Node1] dd if=/dev/urandom of=/gpfs/fs1/demo_mu/file5 bs=256K count=20
20+0 records in
20+0 records out
5242880 bytes (5.2 MB, 5.0 MiB) copied, 0.0268569 s, 195 MB/s
Node1]
```

3. List the files in MU fileset.

```
Node1] ls -shl /gpfs/fs1/demo_mu/
total 14M
1.0M -rw-r--r-- 1 root root 1.0M Jan 12 03:28 file1
2.0M -rw-r--r-- 1 root root 2.0M Jan 12 03:28 file2
3.0M -rw-r--r-- 1 root root 3.0M Jan 12 03:29 file3
4.0M -rw-r--r-- 1 root root 4.0M Jan 12 04:30 file4
4.0M -rw-r--r-- 1 root root 5.0M Jan 12 04:30 file5
Node1]
```

4. Now upload the specific files to COS by using upload command object-list.

- a. Create an object list. The files determined to upload are file1, file3, and file5 as shown in the following sample:

```
Node1] cat /root/objectlist1
/gpfs/fs1/demo_mu/file1
/gpfs/fs1/demo_mu/file3
/gpfs/fs1/demo_mu/file5
Node1]
```

- b. Issue upload command by using the following object list:

```
Node1] mmafmcosctl fs1 demo_mu /gpfs/fs1/demo_mu/ upload --object-list /root/
objectlist1
 Queued (Total) Failed TotalData
 (approx in Bytes)
 3 (3) 0 9437184
```

Object Upload successfully queued at the gateway.

- c. Issue the following command to check the upload stats:

```
Node1] mmafmcosctl fs1 demo_mu --upload-stats
```

```
mmafmcosctl: Statistics of last or currently running upload are as follows:
Fileset Name Pending Failed Total Total data(Bytes)
Throughput(KB/s)
----- ----- ----- -----
demo_mu 0 0 3 9437184
```

```
0
Node1 12Jan04:58:46 4]
```

5. Check that the files are pushed to COS by using COS CLI. The following files are transferred to COS by using COS command-line tool as shown in the sample output:

```
[2022-01-12 04:58:16 EST] 1.0MiB file1
[2022-01-12 04:58:16 EST] 3.0MiB file3
[2022-01-12 04:58:16 EST] 5.0MiB file5
```

### Example 2: Download

#### Note:

- Manual updates mode fileset does not expect file data changes on the cloud object storage. When the file is downloaded from COS the file is marked as cached and then the file is not downloaded again.
- For downloading files from COS to MU mode, --object-list must be used. Download --all option is not relevant to MU mode as this mode is not aware of the data that are created on the COS by the third-party services.

For download example, a MU fileset relation is created from existing bucket that contains existing data.

Bucket name - download-bucket

```
S3 commandline] ls ibmcos/download-bucket
[2022-01-12 05:32:44 EST] 1.0MiB file1
[2022-01-12 05:32:52 EST] 2.0MiB file2
[2022-01-12 05:33:00 EST] 3.0MiB file3
[2022-01-12 05:33:07 EST] 4.0MiB file4
[2022-01-12 05:33:13 EST] 5.0MiB file5
S3 commandline]
```

1. Create MU fileset with existing bucket.

- a. Setup keys for this bucket.

```
node1] mmafmcoskeys download-bucket:s3.us-east.cloud-object-storage.appdomain.cloud
set 7aff636483224d5keyf0c39 8268e298b91dkeyb8cb61e215bc288
078418743fe7e6ad
node1]
```

- b. Create MU mode fileset.

```
node1] mmafmcosconfig fs1 download-mu --endpoint http://s3.us-east.cloud-object-
storage.appdomain.cloud --bucket download-bucket --object-fs --mode mu
```

2. List files in MU fileset.

```
node1] ls -lsh /gpfs/fs1/download-mu/
total 0
node1]
```

3. Create an object list referencing from which files to download from COS.

```
node1] cat /root/objectlist1
/gpfs/fs1/download-mu/file1
/gpfs/fs1/download-mu/file3
/gpfs/fs1/download-mu/file5
node1 12Jan05:42:58 4]
```

4. Download the files by using objectlist.

```
node1] mmafmcosctl fs1 download-mu /gpfs/fs1/download-mu/ download --object-list /root/
objectlist1
Queued (Total) Failed AlreadyCached TotalData
```

```

 3 (3) 0 0 (approx in Bytes)
Object Download successfully queued at the gateway.
node1]

```

5. Check download stats.

```

node1] mmafmcosctl fs1 download-mu --download-stats
mmafmcosctl: Statistics of last or currently running download are as follows:
Fileset Name Pending Failed Already cached Total Total
data(Bytes) Throughput(KB/s)

download-mu 0 0 0 3
9437184 0

```

6. Check that these files are downloaded in the MU mode fileset.

```

node1] ls -lsh /gpfs/fs1/download-mu/
total 9.0M
1.0M -rwxrwx--- 1 root root 1.0M Jan 12 2022 file1
3.0M -rwxrwx--- 1 root root 3.0M Jan 12 2022 file3
5.0M -rwxrwx--- 1 root root 5.0M Jan 12 2022 file5

```

## AFM to COS upload and download statistics

You can get an ongoing statistics report when AFM to cloud object storage upload and download task is in progress.

AFM to cloud object storage fileset modes have upload and download features to support upload and download files to and from cloud object storage. AFM to cloud object storage filesets can be used in different modes of operations where manual upload and download of files is needed. After the files are uploaded or downloaded from COS **mmafmcosctl** command with **-upload-stats** or **-download-stats** can be used to see the progress of upload or download task respectively. The output of **mmafmcosctl** command shows number of files uploaded or downloaded as well as total data in bytes and its throughput.

**Note:**

In case of failed file download or upload, administrators can run **mmafmctl** **checkUncached** and **checkDirty** command to identify the files to download and upload again, respectively.

### Examples of upload and download statistics:

#### Upload statistics

1. Check the fileset.

```

Filesets in file system 'fs1':
Name Status Path
afmTarget
root Linked /gpfs/fs1
--
demo1bucket Linked /gpfs/fs1/demo1bucket
 http://
s3.us-east.cloud-object-storage.appdomain.cloud:80/demo1bucket

```

2. Create three files in fileset.

```

Node1] dd if=/dev/urandom of=/gpfs/fs1/demo1bucket/file1 bs=256K count=40
40+0 records in
40+0 records out
10485760 bytes (10 MB, 10 MiB) copied, 0.0499466 s, 210 MB/s
Node1] dd if=/dev/urandom of=/gpfs/fs1/demo1bucket/file2 bs=256K count=40
40+0 records in
40+0 records out
10485760 bytes (10 MB, 10 MiB) copied, 0.0494641 s, 212 MB/s
Node1] dd if=/dev/urandom of=/gpfs/fs1/demo1bucket/file3 bs=256K count=40
40+0 records in
40+0 records out

```

```
10485760 bytes (10 MB, 10 MiB) copied, 0.0480755 s, 218 MB/s
Node1]
```

3. Create uploadobjectlist file.

```
Node1] cat /root/uploadobjectlist
/gpfs/fs1/demo1bucket/file1
/gpfs/fs1/demo1bucket/file2
/gpfs/fs1/demo1bucket/file3
Node1]
```

4. Upload these objects by using **mmafmcosctl** command.

```
Node1] mmafmcosctl fs1 demo1bucket /gpfs/fs1/demo1bucket/ upload --object-list /root/
uploadobjectlist
 Queued (Total) Failed TotalData
 (approx in Bytes)
 3 (3) 0 31457280

Object Upload successfully queued at the gateway.
```

5. check the upload progress.

```
Node1] mmafmcosctl fs1 demo1bucket --upload-stats
mmafmcosctl: Statistics of last or currently running upload are as follows:
Fileset Name Pending Failed Total Total data(Bytes)
Throughput(KB/s)

----- -----
demo1bucket 2 0 3 10485760
5120
Node1

Node1] mmafmcosctl fs1 demo1bucket --upload-stats
mmafmcosctl: Statistics of last or currently running upload are as follows:
Fileset Name Pending Failed Total Total data(Bytes)
Throughput(KB/s)

----- -----
demo1bucket 0 0 3 31457280 0
```

## Download statistics

1. Files newfile1-3 are uploaded to COS.

```
COScmline] ls ibmcos/demo1bucket
[2022-01-21 06:15:40 EST] 10MiB file1
[2022-01-21 05:50:04 EST] 10MiB file2
[2022-01-21 05:50:01 EST] 10MiB file3
c7f2n03 21Jan06:43:00 0]
```

2. Create an Object list for download.

```
Node1] cat /root/downloadobjectlist
/gpfs/fs1/demo1bucket/newfile1
/gpfs/fs1/demo1bucket/newfile2
/gpfs/fs1/demo1bucket/newfile3
Node1]
```

3. Download the object.

```
Node1] mmafmcosctl fs1 demo1bucket /gpfs/fs1/demo1bucket/ download --object-list /root/
downloadobjectlist
 Queued (Total) Failed AlreadyCached TotalData
 (approx in Bytes)
 3 (3) 0 0 31457280

Object Download successfully queued at the gateway.
Node1 :59:52 2]
```

4. Issue the following command to track download progress:

```
Node1 :59:53 2] mmafmcosctl fs1 demo1bucket --download-stats
mmafmcosctl: Statistics of last or currently running download are as follows:
Fileset Name Pending Failed Already cached Total
data(Bytes) Throughput(KB/s)
```

|             |   |   |   |   |
|-------------|---|---|---|---|
| demo1bucket | 0 | 0 | 0 | 3 |
| 31457280    | 0 | 0 | 0 | 3 |

5. List the files in the fileset to confirm the download.

```
Node1] ls /gpfs/fs1/demo1bucket/
file1 file2 file3 newfile1 newfile2 newfile3
```

## Synchronization of AFM to cloud object storage data to the bucket by using prefix

Prefixes can be used to organize AFM to cloud object storage fileset data inside a bucket. You can upload and download fileset data to the prefix inside the target bucket by creating an AFM to cloud object storage fileset by using user-defined prefix.

To use prefix, you must specify `--prefix <prefix>` parameter while creating AFM to cloud object storage fileset by using `mmafmcosconfig` command. After an AFM fileset is created by using prefix, it synchronizes data directly to the prefix every time. AFM appends the root of the bucket with the prefix created and uses this prefix as the target path of the AFM to cloud object storage fileset and sync all the operations to this path.

The prefix is useful when you do not have access to the root level of the bucket. You can use prefix to upload or download fileset data inside that bucket prefix path.

### Example: Creating fileset by using prefix to sync data to the bucket

The following example shows how to create a fileset by using prefix to sync AFM to cloud object storage fileset data to the bucket prefix.

1. Obtain access key, secret key, region, and URL for a bucket similar to the following as shown here.

```
AccessKey = key1234567890
SecretKey = key1234567890
region = us-west-1
url = s3.amazonaws.com
```

2. After the step #1 is completed, issue the following command to add keys with AFM to use it for data synchronization:

```
mmafmcoskeys bkt1:region@endpoint set AccessKey SecretKey
```

Issue the following command to retrieve the keys:

```
mmafmcoskeys bkt1:region@endpoint get
```

3. Issue the following command to create an AFM to cloud object storage fileset by using prefix. The following command creates a prefix inside a bucket and uses this prefix as target root path of the bucket.

```
mmafmcosconfig fs1 afmbktprefix1 --endpoint https://region@endpoint --object-fs --xattr --
prefix dir1 --bucket bkt1 --acl s --mode sw
```

The preceding example creates a prefix to the existing bucket `bkt1`, then creates an AFM to cloud object storage fileset and append the prefix to the target of the fileset. This prefix becomes the root level of the bucket and all the data is synchronized to this location.

# Migration of a transparent cloud tiering-enabled IBM Storage Scale fileset or file system to an AFM to cloud object storage fileset in the manual update mode

---

AFM can perform inline migration of a transparent cloud tiering (TCT) enabled IBM Storage Scale-independent fileset or file system by promoting it to an AFM to cloud object storage fileset in the manual update (MU) mode. During the migration, file data is not recalled from a cloud tier to a cache fileset. AFM replaces TCT attributes from all files to AFM-specific attributes after the conversion.

When TCT is migrating data and metadata of a file to a cloud tier, local data blocks of the file are evicted from the IBM Storage Scale file system. TCT can recall this file from the cloud so that the data blocks are available on both sides.

## States of files in a TCT-enabled fileset

In a TCT-enabled fileset, files can have the following states:

### Resident

File data is local to a file system and is not migrated to the cloud.

### Non-Resident

File data is not local to a file system (evicted) and is migrated to the cloud.

### Co-Resident

File data is recalled from the cloud, and file data is local to a file system and both side of the cloud.

AFM processes tiered files in the Non-Resident or Co-Resident state to the cloud object storage. Also, it converts them to an AFM migrated file by removing TCT attributes and replaces TCT attributes with AFM attributes. An existing TCT-enabled fileset or a file system can be promoted to an MU-mode fileset by issuing the **mmafmcosconfig --convert** command.

During the promotion, AFM queries all the files in the Co-Resident or Non-Resident state in a TCT-enabled fileset by running a policy. It lists the list-file and processes it.

- AFM removes TCT information on the cloud location, and replaces it with AFM style object information (actual file path).
- AFM removes TCT attributes from the file on the cache side and sets AFM attributes.
- Synchronizes both cache and cloud-side files data and metadata.
- If the file is in the Non-resident or Co-resident state, AFM keeps data without changing it. For a Co-resident state file, you need not to recall file data from the cloud after the promotion. After the promotion is completed, you can perform all operations seamlessly on the AFM to cloud object storage fileset in the MU mode such as upload, download, or evict.

Before the promotion of a TCT fileset, **mmlsattr -X -n <AttributeName> <file>** shows the following TCT attribute. This attribute is removed after the conversion.

```
mmlsattr -X -n dmapi.MCEA /gpfs/fs1/tct3/f11.txt
```

A sample output is as follows:

```
file name: /gpfs/fs1/tct3/f11.txt
```

Files in the Non-resident or Co-resident state in a TCT fileset show the following information on the cloud object storage:

```
002FC5684DD239F6.41DBE301648AD5F2.66DD8A640A006468.0000000000000000.3273D35B.0000000000140134/00
00000000000001.DATA
```

```
002FC5684DD239F6.41DBE301648AD5F2.66DD8A640A006468.0000000000000000.3273D35B.0000000000140134/00
00000000000001.META
```

After the promotion of a TCT-enabled fileset to an AFM MU fileset, AFM removes TCT style format and replaces it with an AFM supported bucket or object style format.

## Migration considerations

- Migration of a TCT-enabled fileset or file system is only for the tiering cloud-service-type. It is not supported for sharing Cloud-Service-Type.
- Before conversion, ensure that a TCT fileset does not have any file or data structure that an AFM to cloud object storage fileset in the MU mode does not support. AFM to cloud object storage limitations such as dependent fileset, file clone not supported. For more information about the AFM limitations, see *AFM to cloud object storage limitations* in the *IBM Storage Scale: Concepts, Planning, and Installation Guide*.
- AFM converts metadata of only TCT-enabled files to AFM supported file metadata because TCT supports tiering of only files and not directories. AFM does not synchronize the directory metadata.
- Before promotion, directories with metadata set are available only at the fileset site. If you want to upload directory metadata to the cloud tier, promote it by using the --directory-object option and upload it by issuing the **mmafmcosctl upload** command after the promotion.
- Symlinks, which were migrated to the cloud tier, are not supported during the migration. Upload symlinks manually after the promotion by issuing the **mmafmcosctl upload** command.
- All resident files, such as files that TCT did not tier to a cloud, remain only at an IBM Storage Scale location. After the promotion, you can upload these files by using the **mmafmcosctl upload** command.

## Promoting a TCT-enabled fileset to an AFM to cloud object storage fileset in the manual update mode

After TCT is enabled on a fileset, promote this fileset to an AFM to cloud object storage fileset in the manual update (MU) mode.

1. Identify a TCT-enabled fileset that needs to be promoted to an AFM to cloud object storage in the MU mode. Ensure that this fileset is in a good state and in-sync with the cloud tiering.

```
mmfslist fs1 tctfset1 -L
```

2. Collect the end point URL.

```
mmccloudgateway cloudStorageAccessPoint list|grep url
```

Ensure that the URL is the same when configuring the TCT on a fileset.

A sample output is as follows:

```
url : http://s3.amazonaws.com
```

3. Register an access key and a secret key for a bucket, which was defined as a data container while configuring the TCT tiering. Keys can be added to AFM issuing the following command:

```
mmafmcoskeys bkt1:s3.amazonaws.com set akey skey
```

Here, the keys assigned to a cloud bucket must be the same as specified in the **mmccloudgateway data-container** command.

4. Stop the migration activities during the promotion.

5. Promote the TCT-enabled fileset to an AFM to cloud object storage fileset in the MU mode.

```
mmafmcosconfig fs1 tctfset1 --endpoint https://s3.amazonaws.com --object-fs --bucket bkt1 --directory-object --acl --convert --xattr --mode mu
```

6. After the TCT-enabled independent fileset is converted to an AFM to cloud object storage fileset in the MU mode, issue the following command:

```
mmafmcosctl fs1 tctfset1 /gpfs/fs1/tctfset1 checkTCT
```

A list of generated files, which AFM needs to promote, is displayed.

```
List of files tiered using the TCT for the fileset tctfset1 - /gpfs/fs1/tctfset1/.mcstore/.mceaEnabledFiles.list.mceaFiles.3290537
```

7. Perform a lookup on the generated tiered files by issuing the following command. The TCT style object information is replaced with the AFM style objects information in the bucket. Therefore, files are not recalled.

```
cat /gpfs/fs1/tctfset1/.mcstore/.mceaEnabledFiles.list.mceaFiles.3290537 |xargs ls >/dev/null
```

8. When the fileset state becomes Active, ensure that the AFM attributes are enabled on both cache and cloud tier and the TCT attributes are removed from each file.

```
mmafmctl fs1 getstate -j tctfset1
```

A sample output is as follows:

| Fileset Name | Fileset Target                  | Cache State | Gateway Node | Queue Length | Queue numExec |
|--------------|---------------------------------|-------------|--------------|--------------|---------------|
| tctfset1     | http://s3.amazonaws.com:80/bkt1 | Active      |              |              | afm-rhel91-1  |
| 0            | 23                              |             |              |              |               |

9. Validate AFM attributes of the cache fileset.

```
mmllsattr -L -d /gpfs/fs1/tctfset1/file1
```

A sample output is as follows:

```
file name: /gpfs/fs1/tct3/f11.txt
```

10. Verify that all TCT information from the cloud tiering is replaced by AFM attributes by querying the bucket. After the promotion, the following objects must be listed:

```
[2023-10-11 12:42:40 PDT] 1.0MiB file1
[2023-10-11 12:42:41 PDT] 1.0MiB file2
[2023-10-11 12:42:42 PDT] 1.0MiB file3
[2023-10-11 12:42:42 PDT] 1.0MiB file4
[2023-10-18 22:06:15 PDT] 0B .afm/
[2023-10-18 22:06:15 PDT] 0B dir1/
[2023-10-18 22:06:15 PDT] 0B dir2/
[2023-10-18 22:06:15 PDT] 0B dir3/
[2023-10-18 22:06:15 PDT] 0B dir4/
```

11. After all operations are finished, issue the following command to list remaining file, if any. For more information, see Step 6.

```
mmafmcosctl fs1 tctfset1 /gpfs/fs1/tctfset1 checkTCT
```

If any file is found, perform the lookup operation again. For more information, see Step 7.

```
cat /gpfs/fs1/tctfset1/.mcstore/.mceaEnabledFiles.list.mceaFiles.3290537 |xargs ls > /dev/null
```

12. Verify that state of the AFM fileset is shown as Active.
13. After all data is validated on both cache and cloud, move the .mcstore directory in the fileset path. This directory can be deleted later.
14. After you corroborate the successful migration from TCT to AFM, you must disable TCT by following the procedure detailed in the *Permanently unistall Cloud services and clean up the environment* section in *IBM Storage Scale: Administration Guide*.

Perform all AFM operations such as upload or download as required.

## Promoting a TCT-enabled file system to an AFM to cloud object storage fileset in the manual update mode

After TCT is enabled on a file system, promote this file system to an AFM to cloud object storage fileset in the manual update (MU) mode.

1. Identify a TCT-enabled IBM Storage Scale file system, which needs to be promoted to an AFM to cloud object storage fileset in the MU mode.

```
#mmlsfileset fs1 -j root
```

- Ensure that this file system is in a good state and in-sync with the cloud tiering.
- Ensure that the file system does not have any independent or dependent fileset on it.

2. Collect the end point URL.

```
mmcloudgateway cloudStorageAccessPoint list | grep url
```

Ensure that the URL is the same when configuring the TCT on a file system.

A sample output is as follows:

```
url : http://s3.amazonaws.com
```

3. Register an access key and a secret key for a bucket, which was defined as a data container when configuring the TCT tiering. Keys can be added to AFM issuing the following command:

```
mmafmcoskeys bkt1:s3.amazonaws.com set akey skey
```

Here, the key of a bucket must be the same as specified to the TCT.

4. Stop the migration activities during the promotion.
5. Promote the TCT-enabled file system to an AFM to cloud object storage file system in the MU mode.

```
mmafmcosconfig fs1 root --endpoint https://s3.amazonaws.com --object-fs --bucket bkt1 --directory-object --acls --convert --xattr --mode mu
```

**Important:** During the conversion of an IBM Storage Scale file system to an AFM to cloud object storage fileset in the MU mode, provide root as name of the fileset.

6. After the TCT-enabled independent fileset is converted to an AFM to cloud object storage fileset in the MU mode, issue the following command:

```
mmafmcosctl fs1 root /gpfs/fs1/ checkTCT
```

A list of generated files, which AFM needs to promote, is displayed.

```
List of files tiered using the TCT for the fileset root - /gpfs/fs1/.mcstore/.mceaEnabledFiles.list.mceaFiles.3520029.
```

7. Perform a lookup on the generated tiered files by issuing the following command. The TCT style object information is replaced with the AFM style objects information in the bucket. Therefore, files are not recalled.

```
cat /gpfs/fs1/.mcstore/.mceaEnabledFiles.list.mceaFiles.3520029 |xargs ls > /dev/null
```

8. After the fileset state becomes Active, ensure that the AFM attributes are enabled on both cache and cloud tier and the TCT attributes are removed from each file.

```
mmalfmctl fs1 getstate -j root
```

A sample output is as follows:

| Fileset Name | Fileset Target | Cache State | Gateway Node | Queue Length | Queue numExec |
|--------------|----------------|-------------|--------------|--------------|---------------|
| -----        | -----          | -----       | -----        | -----        | -----         |

|      |                                         |        |              |
|------|-----------------------------------------|--------|--------------|
| root | http://s3.amazonaws.com:80/bkt1<br>0 23 | Active | afm-rhel91-1 |
|------|-----------------------------------------|--------|--------------|

9. Validate AFM attributes of the cache fileset.

```
mmlsattr -L -d /gpfs/fs1/file1
```

A sample output is as follows:

```
file name: /gpfs/fs1/tct3/f11.txt
```

10. Verify that all TCT information from the cloud tiering is replaced by AFM attributes by querying the bucket. After the promotion, the following objects must be listed:

```
[2023-10-11 12:42:40 PDT] 1.0MiB file1
[2023-10-11 12:42:41 PDT] 1.0MiB file2
[2023-10-11 12:42:42 PDT] 1.0MiB file3
[2023-10-11 12:42:42 PDT] 1.0MiB file4
[2023-10-18 22:06:15 PDT] 0B .afm/
[2023-10-18 22:06:15 PDT] 0B dir1/
[2023-10-18 22:06:15 PDT] 0B dir2/
[2023-10-18 22:06:15 PDT] 0B dir3/
[2023-10-18 22:06:15 PDT] 0B dir4/
```

11. After all operations are finished, issue the following command to list remaining files, if any. For more information, see Step [6](#).

```
mmamfcosctl fs1 root /gpfs/fs1/ checkTCT
```

If any file is found, perform the lookup operation again. For more information, see Step [7](#).

```
cat /gpfs/fs1/tctfset1/.mcstore/.mceaEnabledFiles.list.mceaFiles.3290537 |xargs ls > /dev/null
```

12. Verify that the state of the AFM fileset is shown as Active. If any MCEA enabled files are not on the cache side (you can again check by issuing the **# mmamfcosctl fs1 tctfset1 /gpfs/fs1/tctfset1 checkTCT**), move out of the .mcstore directory in the fileset path to the other backup location. This directory must not exist in the fileset before running any upload operations.
13. The cloud bucket might still have CDIR and JDLT TCT directories residue. After the fileset state is validated, remove these TCT directories also.
14. Remove the cloud container that is provided for this fileset when the following command was issued:

```
mmccloudgateway containerPairSet --meta-container
```

15. After you corroborate the successful migration from TCT to AFM, you must disable TCT by following the procedure detailed in the *Permanently unistall Cloud services and clean up the environment* section in *IBM Storage Scale: Administration Guide*.

Perform all AFM operations such as upload or download as required.

## Converting an AFM to cloud object storage fileset supporting Azure Blob storage by using a MinIO gateway to native Azure Blob storage

In earlier releases of IBM Storage Scale 5.1.9, the AFM to cloud object storage for Azure Blob is executed by using a MinIO gateway. From IBM Storage Scale 5.1.9, an AFM to cloud object storage fileset can connect to Azure Blob natively without any intermediate gateway or application.

To change the AFM to cloud object storage filesets to support Azure Blob naively, complete the following steps:

1. After the storage accounts and keys are set up from an Azure Blob storage, copy the endpoint URL that you want to add as a target and failover to it by using the following command:

```
mmafmctl fs1 failover --target-only
```

A sample output is as follows:

```
https://afmtest.blob.core.windows.net/container1
```

where:

**afmtest**

Is a storage account.

**container1**

Is the name of a container.

2. Stop the IOs on the fileset and change the **afmObjectAZ** fileset parameter to yes.

```
mmchfileset fs1 test -p afmObjectAZ=yes
```

A sample output is as follows:

```
Fileset test changed.
```

3. Change the target.

```
mmafmctl fs1 failover -j test --new-target https://afmtest.blob.core.windows.net/container1 --target-only
```

A sample output is as follows:

```
mmafmctl: Performing failover to https://afmtest.blob.core.windows.net/container1
Fileset test changed.
```

4. Create some data and check whether the queue is active and data is synchronized to the Azure Blob storage.

```
echo 12345 >> /gpfs/fs1/test/fffff
c7f2n02 01Nov10:40:47 2] getstate
```

A sample output is as follows:

| Fileset | Name | Fileset | Target                                               | Cache | State  | Gateway | Node | Queue | Length | Queue | numExec |
|---------|------|---------|------------------------------------------------------|-------|--------|---------|------|-------|--------|-------|---------|
| test    |      |         | https://afmtest.blob.core.windows.net:443/container1 |       | Active | c7f2n02 | 0    | 4     |        |       |         |

# Chapter 60. Highly available write cache (HAWC)

Highly available write cache (HAWC) reduces the latency of small write requests by initially hardening data in a non-volatile fast storage device prior to writing it back to the backend storage system.

## Overview and benefits

Current disk drive systems are optimized for large streaming writes, but many workloads such as VMs and databases consist of many small write requests, which do not perform well with disk drive systems. To improve the performance of small writes, storage controllers buffer write requests in non-volatile memory before writing them to storage. This works well for some workloads, but the amount of NVRAM is typically quite small and can therefore not scale to large workloads.

The goal of HAWC is to improve the efficiency of small write requests by absorbing them in any nonvolatile fast storage device such as SSDs, Flash-backed DIMMs, or Flash DIMMs. Once the dirty data is hardened, GPFS can immediately respond to the application write request, greatly reducing write latency. GPFS can then flush the dirty data to the backend storage in the background.

By first buffering write requests, HAWC allows small writes to be gathered into larger chunks in the page pool before they are written back to storage. This has the potential to improve performance as well by increasing the average amount of data that GPFS writes back to disk at a time.

Further, when GPFS writes a data range smaller than a full block size to a block for the first time, the block must first be fully initialized. Without HAWC, GPFS does this by writing zeroes to the block at the time of the first write request. This increases the write latency since a small write request was converted into a large write request (for example, a 4K write request turns into a 1MB write request). With HAWC, this initialization can be delayed until after GPFS responds to the write request, or simply avoided altogether if the application subsequently writes the entire block.

To buffer the dirty data, HAWC hardens write data in the GPFS recovery log. This means that with HAWC, the recovery log must be stored on a fast storage device because if the storage device on which the recovery log resides is the same as the data device, HAWC will decrease performance by writing data twice to the same device. By hardening data in the recovery log, all incoming requests are transformed into sequential operations to the log. In addition, it is important to note that applications never read data from the recovery log, since all data that is hardened in the recovery log is always kept in the page pool. The dirty data in the log is only accessed during file system recovery due to improper shutdown of one or more mounted instances of a GPFS file system.

The maximum size of an individual write that can be placed in HAWC is currently limited to 64KB. This limit has been set for several reasons, including the following:

- The benefit of writing data to fast storage decreases as the request size increases.
- Fast storage is typically limited to a much smaller capacity than disk subsystems.
- Each GPFS recovery log is currently limited to 1GB. Every file system and client pair has a unique recovery log. This means that for each file system, the size of HAWC scales linearly with every additional GPFS client. For example, with 2 file systems and 10 clients, there would be 20 recovery logs used by HAWC to harden data.

Note that combining the use of HAWC with LROC allows GPFS to leverage fast storage on application reads and writes.

## Applications that can benefit from HAWC

Typically, it is recommended to place GPFS metadata in a storage pool consisting of fast storage devices such as SSDs. Storing GPFS recovery logs in fast storage improves the performance of metadata-intensive workloads where the recovery log is heavily used and when GPFS is configured to replicate data.

With HAWC, storing the recovery log in fast storage has the added benefit that workloads that experience bursts of small and synchronous write requests (no matter if they are random or sequential) will also be hardened in the fast storage. Well-known applications that exhibit this type of write behavior include VMs, databases, and log generation.

Since the characteristics of fast storage vary greatly, users should evaluate their application workload with HAWC in their storage configuration to ensure a benefit is achieved. In general, however, speedups should be seen in any environment that either currently lacks fast storage or has very limited (and non-scalable) amounts of fast storage.

## Restrictions and tuning recommendations for HAWC

---

When enabling HAWC, take the following restrictions and tuning recommendations into consideration:

### Ping pong recovery log buffers

Ping pong recovery log buffers should not be enabled when the recovery log is stored on storage devices that can gracefully write data upon power failure. This restriction includes SSDs, NVRAM, storage controllers, and RAID controllers among others.

Ping pong buffers are only needed to avoid data corruption when the recovery log is stored directly on disk. They place log data in two separate locations on disk to avoid loss of that data if a sector becomes unavailable. Writing to two separate locations creates additional overhead that is exacerbated by HAWC due to the large amount of data it places in the recovery log.

In general, it is not recommended to use HAWC with any storage device that requires ping pong buffers to be enabled because it doubles the amount of data that must be written before GPFS can respond to an application write request.

To disable log buffers, run the following command:

```
mmchconfig logPingPongSector=no
```

### Recovery log size

The size of the recovery log defaults to a small value (less than 16 MB), which is not sufficient space to buffer HAWC data. Therefore, it is recommended to increase the size of the log at least to 128 MB or larger (1 GB maximum). However, the effect of a larger recovery log is that upon node failure, more data must be recovered into the storage system, which increases the time that it takes to recover. It is important to take this factor into account because applications will not be able to access data in the file system while recovery is running.

### Encryption

Encrypted data is never stored in the recovery log, but instead follows the pre-GPFS 4.1.0.4 semantics for synchronous writes even if the HAWC threshold is set to a value greater than 0.

### Small files and directory blocks

HAWC does not change the following behaviors:

- write behavior of small files when the data is placed in the inode itself
- write behavior of directory blocks or other metadata

## Using HAWC

---

Learn how to enable HAWC, set up storage for the recovery log, and do administrative tasks.

[“Enabling HAWC” on page 973](#)

[“Setting up the recovery log in fast storage” on page 973](#)

[“Administrative tasks” on page 973](#)

## Enabling HAWC

To enable HAWC, set the write cache threshold for the file system to a value that is a multiple of 4 KB and in the range 4 KB - 64 KB. The following example shows how to set the threshold for an existing file system:

```
mmchfs gpfsA --write-cache-threshold 32K
```

The following example shows how to specify the threshold when you create a new file system:

```
mmcifs /dev/gpfsB -F ./deiskdef2.txt -B1M --write-cache-threshold 32K -T /gpfs/gpfsB
```

After HAWC is enabled, all synchronous write requests less than or equal to the write cache threshold are put into the recovery log. The file system sends a response to the application after it puts the write request in the log. If the size of the synchronous write request is greater than the threshold, the data is written directly to the primary storage system in the usual way.

## Setting up the recovery log in fast storage

Proper storage for the recovery log is important to improve the performance of small synchronous writes and to ensure that written data survives node or disk failures. Two methods are available:

### Method 1: Centralized fast storage

In this method, the recovery log is stored on a centralized fast storage device such as a storage controller with SSDs, a flash system, or an IBM Elastic Storage Server (ESS) with SSDs.

You can use this configuration on any storage that contains the system pool or the system.log pool. The faster that the metadata pool is compared to the data storage, the more using HAWC can help.

### Method 2: Distributed fast storage in client nodes

In this method, the recovery log is stored on IBM Storage Scale client nodes on local fast storage devices, such as SSDs, NVRAM, or other flash devices.

The local device NSDs must be in the system.log storage pool. The system.log storage pool contains only the recovery logs.

It is a good idea to enable at least two replicas of the system.log pool. Local storage in an IBM Storage Scale node is not highly available, because a node failure makes the storage device inaccessible.

Use the mmchfs command with the --log-replicas parameter to specify a replication factor for the system.log pool. This parameter, with the system.log capability, is intended to place log files in a separate pool with replication different from other metadata in the system pool.

You can change log replication dynamically by running the mmchfs command followed by the mmcstripesfs command. However, you can enable log replication only if the file system was created with a number of maximum metadata replicas of 2 or 3. (See the -M option of the mmcifs command.)

## Administrative tasks

Learn how to do the following administrative tasks with HAWC:

### Restriping after you add or remove a disk

As with any other pool, after you add or remove a disk from the system.log pool, run the mmcstripesfs -b command to rebalance the pool.

### Preparing for a node or disk failure in the system.log pool

- If the system.log is replicated, you can run the following command to ensure that data is replicated automatically after a node or disk fails:

```
mmchconfig restripeOnDiskFailure=yes -i
```

- You can run the following command to set how long the file system waits to start a restripe after a node or disk failure:

```
mmchconfig metadataDiskWaitTimeForRecovery=Seconds
```

where *Seconds* is the number of seconds to wait. This setting helps to avoid doing a restripe after a temporary outage such as a node rebooting. The default time is 300 seconds.

### **Adding HAWC to an existing file system**

Follow these steps:

1. If the metadata pool is not on a fast storage device, migrate the pool to a fast storage device. For more information, see “[Managing storage pools](#)” on page 527.
2. Increase the size of the recovery log to at least 128 MB. Enter the following command:

```
mmchfs Device -L LogFileSize
```

where *LogFileSize* is the size of the recovery log. For more information, see the topic *mmchfs command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

3. Enable HAWC by setting the write cache threshold, as described earlier in this topic.

# Chapter 61. Local read-only cache

Many applications benefit greatly from large local caches. Not only is the data available with very low latency, but the cache hit serves to reduce the load on the shared network and on the backend storage itself, thus benefiting all nodes, even those without large caches.

Local solid-state disks (SSDs) provide an economical way to create very large caches. The SSD cache serves as an extension to the local buffer pool. As user data or metadata is evicted from the buffer pool in memory, it can be stored in the local cache. A subsequent access retrieves the data from the local cache, rather than from the home location. The data stored in the local cache, like data that is stored in memory, remains consistent. If a conflicting access occurs, the data is invalidated from all caches. In a like manner, if a node is restarted, all data stored in the cache is discarded.

In theory, any data or metadata can be stored in the local SSD cache, but the cache works best for small random reads where latency is a primary concern. Since the local cache typically offers less bandwidth than the backend storage, it might be unsuitable for large sequential reads. The configuration options provide controls over what is stored in the cache. The default settings are targeted at small random I/O.

The local read-only cache (LROC) function is disabled by default. To enable it, the administrator must define an NSD for an LROC device. The LROC device is expected to be a solid-state disk (SSD) accessible via SCSI. The device is defined as a standard NSD by `mmcrlnsd`, but the `DiskUsage` is set to `localCache`. The NSD must have a primary server and is not allowed to have other servers. The primary server must be the node where the physical LROC device is installed. The device is *not* exported to other nodes in the cluster. The storage pool and failure group that is defined for the NSD are ignored and must be set to null. The `mmcrlnsd` command writes a unique NSD volume ID onto the device. LROC devices are not tied to filesystems and therefore NSD limits are not affected by LROC devices.

The minimum size of a local read-only cache device is 4 GB. The maximum size of a local read-only cache device is 4 TB. The local read-only cache requires memory equal to 1% of the capacity of the LROC device.

Once the LROC device is defined, the daemon code at the primary server node is automatically told to do device discovery. The daemon detects that `localCache` is defined for its use and determines the mapping to the local device. The daemon then informs the LROC code to begin using the device for caching. Currently, there is a limit of four `localCache` devices per node. Note that the daemon code does not need to be restarted to begin using the cache.

The LROC device can be deleted by using the `mmdelnsd` command. Both `mmcrlnsd` and `mmdelnsd` can be issued while the daemon is running with file systems mounted and online. The call to delete the NSD first informs the daemon that the device is being deleted, which removes it from the list of active LROC devices. Any data that is cached on the device is immediately lost, but data that is cached on other local LROC devices is unaffected. Once the `mmdelnsd` command completes, the underlying SSD can be physically removed from the node.

The NSD name for the LROC device cannot be used in any other GPFS commands, such as `mmcrfs`, `mmadddisk`, `mmrpldisk`, `mmchdisk`, or `mmchnsd`. The device is shown by `mm1snsd` as a `localCache`.

**Note:** To avoid a security exposure, by default IBM Storage Scale does not allow file data from encrypted files, which is held in memory as cleartext, to be copied into an LROC device. However, you can set IBM Storage Scale to allow cleartext from encrypted files to be copied into an LROC device with the following command:

```
mmchconfig lrocEnableStoringClearText=yes
```

You might choose this option if you have configured your system to remove the security exposure.

**Warning:** If you allow cleartext from an encrypted file to be copied into an LROC device, you must take steps to protect the cleartext while it is in LROC storage.

For more information, see the following links:

[“Encryption and a local read-only cache \(LROC\) device” on page 862](#)

The topic *mmchconfig command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

**Note:** To reference data stored in LROC, IBM Storage Scale might require more buffer descriptors than the default number of buffer descriptors on a particular node. The amount of Buffer Descriptors that are desired can be controlled by the **maxBufferDescs** configuration value. This can be set cluster wide or node local. Given that local read-only cache is a node local component, you can run the following command:

```
mmchconfig maxBufferDescs=DesiredNumber -N Node
```

Where *Node* is the node with the local read-only cache device that is attached to configure the desired Buffer Descriptors on that particular node.

The amount of buffer descriptors that are required can vary depending on multiple variables. These variables include the capacity of the LROC devices that are configured and the type of workloads that are running on the node with the LROC devices. In IBM Storage Scale version 5.1.1 or later, warnings are written to the *mmfs.log* if the amount of buffer descriptors that are allocated is deemed too low based on the LROC device capacity. This warning would be displayed in System Health as a TIPS event if System Health is enabled.

Example message in *mmfs.log*:

```
2021-02-18_18:49:37.087-0700: [W] This node has LROC devices with total capacity of 50.0 GB.
Optimal LROC performance requires setting the maxBufferDescs config option.
Based on an assumed 4 MB data block size, the recommended value for maxBufferDescs is 45567 on
this node.
```

For more information, see *Monitoring local read-only cache* in *IBM Storage Scale: Problem Determination Guide*.

# Chapter 62. Miscellaneous advanced administration topics

The following topics provide information about miscellaneous advanced administration tasks:

- [“Changing IP addresses or host names of cluster nodes” on page 977](#)
- [“Enabling a cluster for IPv6” on page 980](#)
- [“Using multiple token servers” on page 981](#)
- [“Exporting file system definitions between clusters” on page 981](#)
- [“IBM Storage Scale port usage” on page 982](#)

## Changing IP addresses or host names of cluster nodes

IBM Storage Scale assumes that the IP addresses and host names of cluster nodes remain constant. In the rare event that a change is necessary, follow the steps in this topic.

Changing the IP addresses or host names of cluster nodes might be necessary either for external reasons or because of an inadvertent action, such as reinstalling a node with a disk image from a different node.

[“First scenario: All the nodes in the cluster are affected” on page 977](#)

[“Second scenario: Some of the nodes in the cluster are affected” on page 978](#)

[“Updating IP addresses or host names in other configurations” on page 979](#)

### Important:

- For both of the scenarios that are described in this topic, a majority of the quorum nodes in the cluster must be up and must be reachable through either the old IP addresses or the new IP addresses.
- For any IP addresses or host names that are changed, be sure to update the IP addresses or host names as needed in other IBM Storage Scale configurations, such as performance monitoring and call home groups. For more information, see the [Updating IP addresses or host names in other configurations](#) procedure.

### First scenario: All the nodes in the cluster are affected

Follow the steps in this subtopic if both of the following conditions are true:

- The IP address or host name of all of the nodes in the cluster has changed.
- One of the following conditions is true:
  - The minimum release level of the cluster is 5.1.0 or later.
  - The minimum release level of the cluster is earlier than 5.1.0, but you can make all of the old host names or IP addresses active on the network.

Otherwise, follow the steps in the next subtopic.

1. Issue the following command to stop IBM Storage Scale on all the nodes in the cluster:

```
mmshutdown -a
```

2. Using the documented procedures for the operating system, add the new host names or IP addresses to the network but do not remove the old ones yet. Do not restart the GPFS daemon on the nodes yet.
3. Follow the documented operating system procedures to make the old host names or IP addresses active on the network without causing network conflicts with the new host names and IP addresses. For example, you might be able to create temporary network aliases for the old IP addresses or host names.

In some cases you might not be able to make the old host names or IP addresses active on the network. For example, if a node has only a single adapter, then it cannot have both the old host name or IP address and the new one active on the network. In such cases, specify the **--force** option when you run the **mmchnode** command in Step 4.

4. Issue the **mmchnode** command to update the administration interface and daemon interface of the nodes whose IP addresses or host names have changed:

**Note:** If the minimum release level of the cluster is earlier than 5.1.0, the daemon interface of a quorum node cannot be changed while CCR is enabled. To resolve this problem, temporarily change the quorum node to a nonquorum node, change the daemon interface of the nonquorum node, and change the nonquorum node back to a quorum node.

- a) Create a specification file for the **mmchnode** command. Add to the file the interface changes that need to be made for each node. For example, the specification file /tmp/specfile lists nodes node-6, node-7, and node-8 and specifies the new daemon interface and the new administrative interface for each node:

```
node-6 --daemon-interface=node-6-2.new-localnet.com --admin-interface=node-6-2.new-localnet.com
node-7 --daemon-interface=node-7-2.new-localnet.com --admin-interface=node-7-2.new-localnet.com
node-8 --daemon-interface=node-8-2.new-localnet.com --admin-interface=node-8-2.new-localnet.com
```

- b) Issue the **mmchnode** command to apply the changes in the specification file:

```
mmchnode --spec-file /tmp/specfile
```

If you could not make all the host names or IP addresses active on the network in Step 3, issue the **mmchnode** command with the **--force** option:

```
mmchnode --spec-file /tmp/specfile --force
```

5. If the IP addresses over which the **subnets** attribute is defined are changed, you must update the **subnets** attribute to include the new addresses. To do so, issue the **mmchconfig** command and set the **subnets** attribute to the new specification.
6. If SMB is enabled, complete the [Step 3 in the Updating IP addresses or host names in other configurations procedure](#) before issuing the following command to start the GPFS daemon on all nodes:

```
mmstartup -a
```

7. Remove the old host names and IP addresses from the network.
8. Be sure to update any other IBM Storage Scale configurations that might contain the old IP addresses or host names. For more information, see the [Updating IP addresses or host names in other configurations procedure](#).

## Second scenario: Some of the nodes in the cluster are affected

Follow the steps in this subtopic if your situation does not match the conditions that are required for the first scenario.

1. Before changing any host names or IP addresses, do the following actions:
  - a) Issue the **mmshutdown** command to stop GPFS on all affected nodes.
  - b) If the host names or IP addresses of the primary or secondary GPFS cluster configuration server nodes must change, use the **mmchcluster** command to specify another node to serve as the primary or secondary GPFS cluster configuration server.
  - c) If the host names or IP addresses of an NSD server node must change, temporarily remove the node from being a server with the **mmchnsd** command. Then, after the node has been added back to the cluster, use the **mmchnsd** command to change the NSDs to their original configuration. Use the **mmllsnsd** command to obtain the NSD server node names.
  - d) If the affected node is a Cluster Export Services (CES) node, you must disable CES on the node. To do so, issue the following command:

```
mmchnode -N <node> --ces-disable
```

**Note:** If you disable all the CES nodes in the cluster, the CES configuration is lost, including the exports. For more information, see [Chapter 45, “Implementing Cluster Export Services,” on page 651](#).

- e) Unless all nodes in the cluster are being deleted, ensure that the **mmdeinode** command is run from a node that remains in the cluster.
  2. Change the node names and IP addresses using the documented procedures for the operating system.
  3. If the IP addresses over which the **subnets** attribute is defined are changed, you must update the **subnets** attribute to include the new addresses. To do so, issue the **mmchconfig** command and set the **subnets** attribute to the new specification.
  4. Issue the **mmaddnode** command to restore the nodes to the GPFS cluster.
  5. If necessary, use the **mmchcluster**, **mmchlicense**, and **mmchnsd** commands to restore the original configuration and the NSD servers.
- Note:** You can use the **mmchnode** command if you need to re-enable CES on the node.
6. If SMB is enabled and an IP is changed on any of the CES nodes, complete the following steps:
    - a) Issue the **mmshutdown** command on all the CES nodes.
    - b) Complete the Step 3 in the [Updating IP addresses or host names in other configurations](#) procedure.
    - c) Issue the **mmstartup** command on all the CES nodes.
  7. Issue the **mmstartup** command to start the GPFS daemon on all nodes that you shut down in the Step 1a.
  8. Be sure to update any other IBM Storage Scale configurations that might contain the old IP addresses or host names. For more information, see the [Updating IP addresses or host names in other configurations](#) procedure.

## Updating IP addresses or host names in other configurations

After you change the IP address or host name of a node, ensure that you update other configurations that might contain the old IP address or host name, such as the following configurations:

- Performance monitoring
- Call home groups
- The CTDB nodes file for CES

1. Update the performance monitoring configuration if needed:

- a) Issue the following command to save the current performance monitoring configuration file into a temporary file:

```
mmperfmon config show --config-file <temporary-file-name>
```

- b) Open the temporary file in a text editor and change any occurrence of an old node name or IP address to the new one.

- c) If you changed a node name or IP address in Step 2, issue the following command to update the performance monitoring configuration:

```
mmperfmon config update --config-file <temporary-file-name>
```

- d) If the performance monitoring configuration is a federation, and affected node is a collector, and you are running a version of IBM Storage Scale that is earlier than 5.0.2, update the names and IP addresses of the peers in the /opt/IBM/zimon/ZIMonCollector.cfg file on all the collector nodes.

2. If the long admin node names (FQDN) of any call home group members are changed, you must delete the affected call home groups and create new ones:

- a) Issue the **mmcallhome group list** command to check the status of nodes that are members of call home groups. If the command displays "-----" instead of the name of a node, then the node is deleted or its FQDN (including the domain) has changed.
  - b) If a node is deleted or its FQDN has changed, delete its call home group and create a new one. For more information, see *mmcallhome command* in the *IBM Storage Scale: Command and Programming Reference Guide*.
3. Update the cluster trivial database (CTDB) nodes file in the clustered configuration repository (CCR) to include new host IPs in the file.
- a) Store the `smb.ctdb.nodes` file from the CCR to a temporary location by issuing the following command:

```
mmccr fget smb.ctdb.nodes /tmp/smb.ctdb.nodes
```
  - b) Change the IPs accordingly in the `/tmp/smb.ctdb.nodes` file. The order of IPs must match the previous order of nodes, and must not be changed.

**Note:** These IPs are host IPs, and must not be CES IPs.
- c) Send the updated `/tmp/smb.ctdb.nodes` file to the CCR by issuing the following command:  

```
mmccr fput smb.ctdb.nodes /tmp/smb.ctdb.nodes
```

## Enabling a cluster for IPv6

For newly created clusters, if any of the specified node interfaces on the `mmcrcluster` command resolves to an IPv6 address, the cluster is automatically enabled for IPv6. For existing IPv4-based clusters, follow the applicable procedure described in this section.

If you are performing the procedure during a scheduled maintenance window and GPFS can be shut down on all of the nodes in the cluster, issue the command:

```
mmchconfig enableIPv6=yes
```

After the command finishes successfully, you can start adding new nodes with IPv6 addresses.

If it is not possible to shut down GPFS on all of the nodes at the same time, issue the command:

```
mmchconfig enableIPv6=prepare
```

The next step is to restart GPFS on each of the nodes so that they can pick up the new configuration setting. This can be done one node at a time when it is convenient. To verify that a particular node has been refreshed, issue:

```
mmdiag --config | grep enableIPv6
```

The reported value should be 1.

Once all of the nodes have been recycled in this manner, issue the command:

```
mmchconfig enableIPv6=commit
```

This command will only succeed when all GPFS daemons have been refreshed. Once this operation succeeds, you can start adding new nodes with IPv6 addresses.

To convert an existing node from an IPv4 to an IPv6 interface, use one of the procedures described in “[Changing IP addresses or host names of cluster nodes](#)” on page 977.

## Using multiple token servers

---

Distributed locking, allowing GPFS to maintain a consistent view of the file system, is implemented using token-based lock management. Associated with every lockable object is a token.

Before a lock on an object can be granted to a thread on a particular node, the lock manager on that node must obtain a token from the token server. The total number of token manager nodes depends on the number of manager nodes defined in the cluster.

When a file system is first mounted, the file system manager is the only token server for the file system. Once the number of external mounts exceeds one, the file system manager appoints all the other manager nodes defined in the cluster to share the token server load. Once the token state has been distributed, it remains distributed until all external mounts have gone away. The only nodes that are eligible to become token manager nodes are those designated as manager nodes.

The number of files for which tokens can be retained on a manager node is restricted by the values of the `maxFilesToCache` and `maxStatCache` configuration parameters of the `mmchconfig` command. Distributing the tokens across multiple token manager nodes allows more tokens to be managed or retained concurrently, improving performance in situations where many lockable objects are accessed concurrently.

**Note:** There is no limit to the number of management server nodes that can be configured as token servers. Increasing the number of token servers improves the distribution of tokens amongst the token management servers and reducing delays that can occur when tokens need to be migrated between token management servers.

## Exporting file system definitions between clusters

---

You can export a GPFS file system definition from one GPFS cluster to another.

To export file system definitions between clusters, follow these steps:

1. Ensure that all disks in all GPFS file systems to be migrated are in working order by issuing the `mmlsdisk` command. Verify that the disk status is ready and availability is up. If not, correct any problems and reissue the `mmlsdisk` command before continuing.
2. Stop all user activity in the file systems.
3. Follow any local administrative backup procedures to provide for protection of your file system data in the event of a failure.
4. Cleanly unmount all affected GPFS file systems. Do not use force unmount.
5. Export the GPFS file system definitions by issuing the `mmexportfs` command. This command creates the configuration output file *ExportDataFile* with all relevant file system and disk information. Retain this file as it is required when issuing the `mmimportfs` command to import your file systems into the new cluster. Depending on whether you are exporting a single file system or all of the file systems in the cluster, issue:

```
mmexportfs fileSystemName -o ExportDataFile
```

or

```
mmexportfs all -o ExportDataFile
```

6. Ensure that the file system disks from the old GPFS cluster are properly connected, and are online and available to be accessed from appropriate nodes of the new GPFS cluster.
7. To complete the movement of your file systems to the new cluster using the configuration file created in Step “5” on page 981, issue one of these commands, depending on whether you are importing a single file system or all of the file systems in the cluster:

```
mmimportfs fileSystemName -i ExportDataFile
```

or

```
mmimportfs all -i ExportDataFile
```

## IBM Storage Scale port usage

The nodes in an IBM Storage Scale cluster communicate with each other using the TCP/IP protocol. The port number used by the main GPFS daemon (**mmfsd**) is controlled with the **tscTcpPort** configuration parameter. The default port number is 1191.

You can specify a different port number by using the **mmchconfig** command:

```
mmchconfig tscTcpPort=PortNumber
```

When the main GPFS daemon (**mmfsd**) is not running, a separate service (**mmsdrserv**) is used to provide access to the configuration data. The port number used for this purpose is controlled with the **mmsdrservPort** (**tscTcpPort**) parameter. The **mmsdrserv** daemon uses the same port number as the one assigned to the main GPFS daemon.

Certain commands (**mmadddisk**, **mmchmgr**, and so on) require an additional socket to be created for the duration of the command. The port numbers assigned to these temporary sockets are controlled with the **tscCmdPortRange** configuration parameter. If an explicit range is not specified, the port number is dynamically assigned by the operating system from the range of ephemeral port numbers. If you want to restrict the range of ports used by IBM Storage Scale commands, use the **mmchconfig** command:

```
mmchconfig tscCmdPortRange=LowNumber-HighNumber
```

In a remote cluster setup, if IBM Storage Scale on the remote cluster is configured to use a port number other than the default, you have to specify the port number to be used with the **mmremotecluster** command:

```
mmremotecluster update ClusterName -n tcpPort=PortNumber,Node,Node...
```

For related information, see the topic “[Firewall recommendations for internal communication among nodes](#)” on page 985.

Table 75 on page 982 provides IBM Storage Scale port usage information:

| Table 75. IBM Storage Scale port usage |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Descriptor                             | Explanation                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Service provider                       | IBM Storage Scale                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Service name                           | mmfsd<br>mmsdrserv                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Port number                            | 1191<br><br>While executing certain commands, IBM Storage Scale may need to create additional sockets whose dynamic port numbers are assigned by the operating system. Such sockets are used by commands to exchange data with GPFS daemons running on other nodes. The port numbers that are used correspond to the ephemeral ports of the operating system.<br><br>To control which ports are used by the commands (so that firewall rules can be written to allow incoming traffic only on those ports), you can restrict the port range to a specific range by setting the <b>tscCmdPortRange</b> configuration variable. |

Table 75. IBM Storage Scale port usage (continued)

| Descriptor                                                                                                                                 | Explanation                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Protocols                                                                                                                                  | TCP/IP                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Source port range                                                                                                                          | The source port range is chosen by the operating system on the client side.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Is the service name/number pair in the default /etc/services file shipped with AIX and Linux distributions?                                | See the <a href="#">IBM Storage Scale FAQ in IBM Documentation</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Is the service name/number pair added to /etc/services by a product?                                                                       | No                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Binaries that listen on the ports                                                                                                          | <pre>/usr/lpp/mmfs/bin/mmfssd /usr/lpp/mmfs/bin/mmsdbserv</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Can the service be configured to use a different port?                                                                                     | <p>Yes. To change the main port used by IBM Storage Scale, enter:</p> <pre>mmchconfig tscTcpPort=PortNumber</pre> <p>To change the range of port numbers used for command execution, use:</p> <pre>mmchconfig tscCmdPortRange=LowNumber-HighNumber</pre> <p>To specify a port number when connecting to remote clusters, use the <b>mmremotecluster</b> command.</p>                                                                                                                                                                                                                                                                      |
| When the daemon starts and its port is already in use (for example, another resource has bound to it already), how does the daemon behave? | <p>The daemon shuts down and tries to start over again. Most GPFS daemon down error messages are in the <b>mmfs.log.previous</b> log for the instance that failed. If the daemon restarted, it generates a new <b>mmfs.log.latest</b> log.</p> <p>Begin problem determination for these errors by examining the operating system error log. IBM Storage Scale records file system or disk failures using the error logging facility provided by the operating system: <b>syslog</b> facility on Linux and <b>errpt</b> facility on AIX.</p> <p>See the <i>IBM Storage Scale: Problem Determination Guide</i> for further information.</p> |
| Is there an administrator interface to query the daemon and have it report its port number?                                                | <p>Yes; issue this command:</p> <pre>mmlsconfig tscTcpPort</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Is the service/port registered with the Internet Assigned Numbers Authority (IANA)?                                                        | <p>Yes</p> <pre>gpfs 1191/tcp General Parallel File System gpfs 1191/udp General Parallel File System # Dave Craft &lt;gpfs@ibm.com&gt; November 2004</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

**Note:** Ports configured for the IBM Storage Scale remote shell command (such as ssh) or the remote file copy command (such as scp) and the ICMP echo command (network ping) also must be unblocked in the firewall for IBM Storage Scale to function properly.

## IBM Storage Scale GUI port usage

You can enable the IBM Storage Scale Management GUI to read new HTTP and HTTPS ports to avoid conflict during the use of third-party applications.

You must create the /etc/scale-gui-configuration directory to enable the IBM Storage Scale Management GUI to read the relevant HTTP and HTTPS ports. The following files provide the port information.

### For HTTP port

```
/etc/scale-gui-configuration/scale_gui_http_port
```

### For HTTPS port

```
/etc/scale-gui-configuration/scale_gui_port
```

## Configuring port numbers

You can change GUI nodes for the entire cluster and update the *GUI\_HTTPS\_PORT* and *GUI\_HTTP\_PORT* variables in the gpfsgui.properties file with the port number.

**Note:** Use the same port number for all GUI nodes in the cluster.

### For HTTPS port

Issue the following command to set a non default port number. For example, 445:

```
mkdir /etc/scale-gui-configuration/
echo "445" > /etc/scale-gui-configuration/scale_gui_port
update GUI_HTTPS_PORT=445 in /usr/lpp/mmfs/gui/conf/gpfsgui.properties
systemctl restart gpfsgui
```

### For HTTP port

Issue the following command to set a non default port number. For example, 445:

```
mkdir /etc/scale-gui-configuration/
echo "445" > /etc/scale-gui-configuration/scale_gui_http_port
update GUI_HTTP_PORT=445 in /usr/lpp/mmfs/gui/conf/gpfsgui.properties
systemctl restart gpfsgui
```

## Securing the IBM Storage Scale system using firewall

The IBM Storage Scale system is an open system where the customer can interact with the system through other third-party interfaces like MMC, web applications, and so on. The customer also has root access to the system just like any Linux server administrator. Firewalls that are associated with open systems are specific to deployments, operating systems, and it varies from customer to customer. It is the responsibility of the system administrator or Lab Service (LBS) to set the firewall accordingly; similar to what Linux distributions do today. This section provides recommendations to set up a firewall to secure the IBM Storage Scale protocol nodes.

| Table 76. Firewall related information           |                                                                                                                                                                                                               |
|--------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Function                                         | Firewall recommendations and considerations                                                                                                                                                                   |
| IBM Storage Scale installation                   | <a href="#">“Firewall recommendations for the IBM Storage Scale installation” on page 985</a>                                                                                                                 |
| Internal communication                           | <a href="#">“Firewall recommendations for internal communication among nodes” on page 985</a><br><br>For detailed information on port usage, see <a href="#">“IBM Storage Scale port usage” on page 982</a> . |
| Protocol access (NFS, SMB, S3, and Swift Object) | <a href="#">“Firewall recommendations for protocol access” on page 987</a>                                                                                                                                    |

*Table 76. Firewall related information (continued)*

| <b>Function</b>                                                | <b>Firewall recommendations and considerations</b>                                                                             |
|----------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| IBM Storage Scale GUI                                          | <a href="#">“Firewall recommendations for IBM Storage Scale GUI” on page 991</a>                                               |
| File encryption with IBM Security Key Lifecycle Manager (SKLM) | <a href="#">“Firewall recommendations for IBM SKLM” on page 992</a>                                                            |
| File encryption with Vormetric Data Security Manager (DSM)     | <a href="#">“Firewall recommendations for Thales Vormetric Data Security Manager (DSM)” on page 993</a>                        |
| Performance monitoring                                         | <a href="#">“Firewall recommendations for Performance Monitoring tool” on page 994</a>                                         |
| Active File Management (AFM)                                   | <a href="#">“Firewall considerations for Active File Management (AFM)” on page 994</a>                                         |
| Transparent cloud tiering                                      | <i>Firewall recommendations for Transparent cloud tiering in IBM Storage Scale: Concepts, Planning, and Installation Guide</i> |
| Remotely mounted file systems                                  | <a href="#">“Firewall considerations for remote mounting of file systems” on page 995</a>                                      |
| IBM Storage Protect with IBM Storage Scale                     | <a href="#">“Firewall recommendations for using IBM Storage Protect with IBM Storage Scale” on page 995</a>                    |
| IBM Spectrum Archive with IBM Storage Scale                    | <a href="#">“Firewall considerations for using IBM Spectrum Archive with IBM Storage Scale” on page 995</a>                    |
| Call home                                                      | <a href="#">“Firewall recommendations for call home” on page 996</a>                                                           |
| Examples of opening firewall ports                             |                                                                                                                                |

## Firewall recommendations for the IBM Storage Scale installation

The installation toolkit uses the following ports during IBM Storage Scale installation.

*Table 77. Recommended port numbers that can be used for installation*

| <b>Port Number</b> | <b>Protocol</b> | <b>Service Name</b> | <b>Components involved in communication</b> |
|--------------------|-----------------|---------------------|---------------------------------------------|
| 10080              | TCP             | Repository          | Intra-cluster and installer server          |

You can get the list of protocol IP addresses by using the `mmlscluster --ces` command. Use the `mmlscluster` command to get the list of all internal IP addresses.

NTP is not necessary but time sync among nodes is highly recommended and it is required for protocol nodes.

## Firewall recommendations for internal communication among nodes

The IBM Storage Scale system uses the following ports for internal communication among various IBM Storage Scale nodes.

**Important:** The ports that you plan to use for IBM Storage Scale internal communication might be blocked by a firewall or for some other reason on some nodes in a cluster. If so, then IBM Storage Scale communication errors will occur and some operations might fail. Therefore it is important to verify that the IBM Storage Scale internal communication ports on each node are accessible from every node in the

cluster, including the node itself. Also, if you plan for nodes in one cluster to mount file systems in another cluster, then it is important to verify that all the IBM Storage Scale ports for internal communication in either cluster are accessible by all the nodes in the other cluster. If not, an attempt by a node in one cluster to mount a file system in another cluster might fail, or nodes in the remote cluster might be expelled.

*Table 78. Recommended port numbers that can be used for internal communication*

| Port Number         | Protocol | Service Name                           | Components that are involved in communication |
|---------------------|----------|----------------------------------------|-----------------------------------------------|
| 1191                | TCP      | GPFS                                   | Intra-cluster                                 |
| 22                  | TCP      | Remote shell command, such as SSH.     | Commands                                      |
| 22                  | TCP      | Remote file copy command, such as SCP. | Commands                                      |
| ---                 | ICMP     | ICMP ECHO (ping).                      | Intra-cluster                                 |
| User-selected range | TCP      | GPFS ephemeral port range              | Intra-cluster                                 |

- The SSH and SCP port 22 is used for command execution and general node-to-node configuration as well as administrative access.
- The GPFS and CCR daemons (**mmfsd** and **mmsdrserv**), by default, listen on port 1191. This port is essential for basic cluster operation. The port can be changed manually by setting the *tscTcpPort* configuration variable with the **mmchconfig tscTcpPort=PortNumber** command.
- The ephemeral port range of the underlying operating system is used when IBM Storage Scale creates additional sockets to exchange data among nodes. This occurs while executing certain commands and this process is dynamic based on the point in time needs of the command as well as other concurrent cluster activities. You can define an ephemeral port range manually by setting the *tscCmdPortRange* configuration variable with the **mmchconfig tscCmdPortRange=LowNumber-HighNumber** command.

If the installation toolkit is used, the ephemeral port range is automatically set to 60000-61000. Firewall ports must be opened according to the defined ephemeral port range. If commands such as **mm1smgr** and **mmcrfs** hang, it indicates that the ephemeral port range is improperly configured.

For related information, see the topic “IBM Storage Scale port usage” on page 982.

The following are the recommendations for securing internal communications among IBM Storage Scale nodes:

- Allow connection only to the GPFS cluster node IPs (internal IPs and protocol node IPs) on port 1191. Block all other external connections on this port. Use the **mmlscluster --ces** command to get the list of protocol node IP and use the **mmlscluster** command to get the list of IPs of internal nodes.
- Allow all external communications request that are coming from the admin or management network and IBM Storage Scale internal IPs on port 22.
- Certain commands such as **mmadddisk**, **mmchmgr**, and so on require an extra socket to be created for the duration of the command. The port numbers that are assigned to these temporary sockets are controlled with the **tscCmdPortRange** configuration parameter. If an explicit range is not specified, the port number is dynamically assigned by the operating system from the range of ephemeral port numbers. It is highly recommended to set the port range. For more information on how to set the port range, see “IBM Storage Scale port usage” on page 982.

## Firewall recommendations for protocol access

It is recommended to use certain port numbers to secure the protocol data transfer.

### Recommendations for NFS access

The following table provides the list of static ports that are used for NFS data I/O.

| Table 79. Recommended port numbers for NFS access |             |                                           |                                                 |
|---------------------------------------------------|-------------|-------------------------------------------|-------------------------------------------------|
| Port Number                                       | Protocol    | Service Name                              | Components that are involved in communication   |
| 2049                                              | TCP and UDP | NFSV4 or NFSV3                            | NFS clients and IBM Storage Scale protocol node |
| 111                                               | TCP and UDP | RPC (required only by NFSV3)              | NFS clients and IBM Storage Scale protocol node |
| User-defined static port                          | TCP and UDP | STATD (required only by NFSV3)            | NFS clients and IBM Storage Scale protocol node |
| User-defined static port                          | TCP and UDP | MNT (required only by NFSV3)              | NFS clients and IBM Storage Scale protocol node |
| User-defined static port                          | TCP and UDP | NLM (required only by NFSV3)              | NFS clients and IBM Storage Scale protocol node |
| User-defined static port                          | TCP and UDP | RQUOTA (required by both NFSV3 and NFSV4) | NFS clients and IBM Storage Scale protocol node |

**Note:** NFSV3 uses the dynamic ports for NLM, MNT, and STATD services. When an NFS server is used with the firewall, these services must be configured with static ports.

The following recommendations are applicable:

- Review your systems /etc/services file in order to select the static ports to use for MNT, NLM, STATD, and RQUOTA services that are required by the NFSV4 server. Do not use a port that is already used by another application. Set the static ports by using the **mmnfs config change** command. Allow TCP and UDP port 2049 to use the protocol node IPs. For example:

```
mmnfs config change MNT_PORT=32767:NLM_PORT=32769:RQUOTA_PORT=32768:STATD_PORT=32765
```

- Allow all external communications on TCP and UDP port 111 by using the protocol node IPs.
- Allow all external communications on the TCP and UDP port that is specified with **mmnfs config change** for MNT and NLM ports.
- Ensure that following steps are done after making any of these changes.
  - Restart NFS after changing these parameters by using the following commands.

```
mmces service stop NFS -a
mmces service start NFS -a
```

- Use **xpcinfo -p** to query the protocol nodes after any port changes to verify that proper ports are in use.
- Remount any existing clients because a port change might have disrupted connections.

## Recommendations for SMB access

Samba uses the following ports for the secure access.

| Table 80. Recommended port numbers for SMB access |          |              |                                                 |
|---------------------------------------------------|----------|--------------|-------------------------------------------------|
| Port Number                                       | Protocol | Service Name | Components that are involved in communication   |
| 445                                               | TCP      | Samba        | SMB clients and IBM Storage Scale protocol node |
| 4379                                              | TCP      | CTDB         | Inter-protocol node                             |

Ø

The following recommendations are applicable for the SMB access:

- Allow the access request that is coming from the data network and admin and management network on port 445 using the protocol node IPs. You can get the list of protocol node IPs by using the **mmlscluster --ces** command.
- Allow connection only to the requests that are coming from the IBM Storage Scale cluster node IPs (internal IPs and protocol node IPs) on port 4379. Block all other external connections on this port. Use the **mmlscluster** command to get the list of cluster node IPs.

## Recommendations for the S3 access

Ports for the S3 access are listed in the following table:

| Table 81. Recommended port numbers for the S3 access |          |              |                                               |
|------------------------------------------------------|----------|--------------|-----------------------------------------------|
| Port number                                          | Protocol | Service name | Components that are involved in communication |
| 6443 (default ENDPOINT_SSL_PORT)                     | TCP      | noobaa       | S3 client and IBM Storage Scale protocol node |
| 6001 (default ENDPOINT_PORT)                         | TCP      | noobaa       | S3 client and IBM Storage Scale protocol node |

The following recommendations are applicable for the S3 access:

- Allow the secure access request that is coming from the S3 client and the protocol node on port 6443 for all HTTPS requests that are using the protocol node IPs. You can get the list of protocol node IPs by using the **mmlscluster --ces** command.
- Allow the access request that is coming from the S3 client and the protocol node on port 6001 for all HTTP requests that are using the protocol node CES IPs. You can get the list of protocol node IPs by using the **mmlscluster --ces** command.

If you want to change the default ports, complete the following steps. Ensure that the new ports, which you chose, are not **Active** in the /etc/services on protocol nodes.

- List the current configuration.

```
mms3 config list
```

- Change the default port for HTTPS, that is, ENDPOINT\_SSL\_PORT.

```
mms3 config change ENDPOINT_SSL_PORT=<port-number>
```

3. Change the default port for HTTP, that is, ENDPOINT\_PORT.

```
mms3 config change ENDPOINT_PORT=<port-number>
```

**Note:** The ALLOW\_HTTP=true configuration parameter must be set to **true** along with HTTP port change for I/O requests to take affect from S3 users.

4. Check whether the ports are changed.

```
netstat -an |grep <port-number>
```

5. Ensure that sysadmin communicate to all S3 user accounts on the changed port change, so that user accounts can send I/O requests appropriately.

## Object port configuration

**Note:** IBM Storage Scale is configured with the ports listed here. Changing ports requires updating configuration files, Keystone endpoint definitions, and SELinux rules. This must be done only after careful planning.

The following table lists the ports configured for object access.

| Port Number | Protocol    | Service Name                                                      | Components that are involved in communication      |
|-------------|-------------|-------------------------------------------------------------------|----------------------------------------------------|
| 8080        | TCP         | Object Storage Proxy                                              | Object clients and IBM Storage Scale protocol node |
| 6200        | TCP         | Object Storage (local account server)                             | Local host                                         |
| 6201        | TCP         | Object Storage (local container server)                           | Local host                                         |
| 6202        | TCP         | Object Storage (local object server)                              | Local host                                         |
| 6203        | TCP         | Object Storage (object server for unified file and object access) | Local host                                         |
| 11211       | TCP and UDP | Memcached (local)                                                 | Local host                                         |

The following ports are configured for securing object access:

- Allow all external communications on TCP port 8080 (Object Storage proxy).
- Allow connection only from the IBM Storage Scale cluster node IPs (internal IPs and protocol node IPs) on ports 6200, 6201, 6202, 6203, and 11211. Block all other external connections on this port.

Shell access by non-root users must be restricted on IBM Storage Scale protocol nodes where the object services are running to prevent unauthorized access to object data.

**Note:** The reason for these restrictions is that because there is no authentication of requests made on ports 6200, 6201, 6202, and 6203, it is critical to ensure that these ports are protected from access by unauthorized clients.

### Port usage for object authentication

You can configure either an external or internal Keystone server to manage the authentication requests. Keystone uses the following ports:

*Table 83. Port numbers for object authentication*

| Port Number | Protocol | Service Name            | Components that are involved in communication                |
|-------------|----------|-------------------------|--------------------------------------------------------------|
| 5000        | TCP      | Keystone Public         | Authentication clients and object clients                    |
| 35357       | TCP      | Keystone Internal/Admin | Authentication and object clients and Keystone administrator |

These ports are applicable only if keystone is hosted internally on the IBM Storage Scale system. The following port usage is applicable:

- Allow all external communication requests that are coming from the admin or management network and IBM Storage Scale internal IPs on port 35357.
- Allow all external communication requests that are coming from clients to IBM Storage Scale for object storage on port 5000. Block all other external connections on this port.

#### Port usage to connect to the Postgres database for object protocol

The Postgres database server for object protocol is configured to use the following port:

*Table 84. Port numbers for Postgres database for object protocol*

| Port Number | Protocol    | Service Name   | Components that are involved in communication |
|-------------|-------------|----------------|-----------------------------------------------|
| 5431        | TCP and UDP | postgresql-obj | Inter-protocol nodes                          |

It is recommended to allow connection only from Cluster node IPs (Internal IPs and Protocol node IPs) on port 5431. Block all other communication requests on this port.

**Note:** The Postgres instance used by the object protocol uses port 5431. This is different from the default port to avoid conflict with other Postgres instances that might be on the system including the instance for IBM Storage Scale GUI.

#### Consolidated list of recommended ports that are used for installation, internal communication, and protocol access

The following table provides a consolidated list of recommended ports and firewall rules.

*Table 85. Consolidated list of recommended ports for different functions*

| Function                      | Dependent network service names | External ports that are used for file and object access | Internal ports that are used for inter-cluster communication | UDP / TCP                      | Nodes for which the rules are applicable |
|-------------------------------|---------------------------------|---------------------------------------------------------|--------------------------------------------------------------|--------------------------------|------------------------------------------|
| Installer                     | Ansible®                        | N/A                                                     | 10080 (repo)                                                 | TCP                            | GPFS server, NSD server, protocol nodes  |
| GPFS (internal communication) | GPFS                            | N/A                                                     | 1191 (GPFS)<br>60000-61000 for tscCmdPortRange<br>22 for SSH | TCP and UDP<br>TCP only for 22 | GPFS server, NSD server, protocol nodes  |

Table 85. Consolidated list of recommended ports for different functions (continued)

| Function | Dependent network service names                      | External ports that are used for file and object access                                                                                                                                                                                                                                                                                                                  | Internal ports that are used for inter-cluster communication                       | UDP / TCP                           | Nodes for which the rules are applicable |
|----------|------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|-------------------------------------|------------------------------------------|
| SMB      | gpfs-smb.service<br>gpfs-ctdb.service<br>rpc.statd   | 445                                                                                                                                                                                                                                                                                                                                                                      | 4379 (CTDB)                                                                        | TCP                                 | Protocol nodes only                      |
| NFS      | gpfs.ganesha.nfsd<br>rpcbind<br>rpc.statd            | 2049 (NFS_PORT - required only by NFSV3)<br><br>111 (RPC - required only by NFSV3)<br><br>32765 (STATD_PORT)<br><br>32767 (MNT_PORT - required only by NFSV3)<br><br>32768 (RQUOTA_PORT - required by both NFSV3 and NFSV4)<br><br>32769 (NLM_PORT - required only by NFSV3)<br><br><b>Note:</b> Make the dynamic ports static with command <b>mmnfs config change</b> . | N/A                                                                                | TCP and UDP                         | Protocol nodes only                      |
| S3       | noobaa.service<br>6443 (default SSL_PORT)            | 6001 (default HTTP PORT)                                                                                                                                                                                                                                                                                                                                                 | N/A                                                                                | TCP                                 | Protocol nodes only                      |
| Object   | swift-proxy-server<br>keystone-all<br>postgresql-obj | 8080 (proxy server)<br>35357 (keystone)<br>5000 (keystone public)                                                                                                                                                                                                                                                                                                        | 5431 (Object Postgres instance)<br>6200-6203 (Object Storage)<br>11211 (Memcached) | TCP<br>TCP and UDP (for 11211 only) | Protocol nodes only                      |

## Firewall recommendations for IBM Storage Scale GUI

Dedicating certain ports for firewalls helps to secure the IBM Storage Scale management GUI.

The following table lists the ports that need to be used to secure GUI.

Table 86. Firewall recommendations for GUI

| Port Number   | Functions                                          | Protocol            |
|---------------|----------------------------------------------------|---------------------|
| 47080         | Management GUI                                     | TCP, localhost only |
| 47443         | Management GUI                                     | TCP, localhost only |
| 80            | Management GUI<br>IBM Storage Scale management API | TCP                 |
| 443           | Management GUI<br>IBM Storage Scale management API | TCP                 |
| 4444          | Management GUI                                     | TCP, localhost only |
| 4739          | Performance monitoring tool                        | TCP and UDP         |
| 9980 and 9981 | Performance monitoring tool                        | TCP                 |

All nodes of the IBM Storage Scale cluster must be able to communicate with the GUI nodes through the ports 80 and 443. If multiple GUI nodes are available in a cluster, the communication among those GUI nodes is carried out through the port 443.

Both the management GUI and IBM Storage Scale management API share the same ports. That is, 80 and 443. However, for APIs, the ports 443 and 80 are internally forwarded to 47443 and 47080 respectively. This is done automatically by an *iptables* rule that is added during the startup of the GUI and is removed when the GUI is being stopped. The update mechanism for iptables can be disabled by setting the variable **UPDATE\_IPTABLES** to *false*, which is stored at: /etc/sysconfig/gpfsgui.

**Note:** The GUI cannot coexist with a web server that uses the same ports. You can change the GUI ports to avoid any conflicts. For more information, see “[IBM Storage Scale GUI port usage](#)” on page 984.

If you are installing GUI on RHEL 9 then you must install *nftables*.

The management GUI uses ZIMon to collect performance data. ZIMon collectors are normally deployed with the management GUI and sometimes on other systems in a federated configuration. Each ZIMon collector uses three ports, which can be configured in ZIMonCollector.cfg. The default ports are 4739, 9980, and 9981. The GUI is sending its queries on the ports 9980, and 9981 and these ports are accessible only from the localhost. For more information on the ports used by the performance monitoring tools, see [“Firewall recommendations for Performance Monitoring tool” on page 994](#).

The port 4444 is accessible only from the localhost.

## Firewall recommendations for IBM SKLM

Read this topic to learn about port access for IBM Security Key Lifecycle Manager (SKLM).

The following table lists the ports for communicating with SKLM. The SKLM ports apply for both IBM Storage Scale file encryption and Transparent Cloud Tiering (TCT).

**Note:** IBM Storage Scale supports IBM Security Guardium Key Lifecycle Manager (GKLM) 4.1.0.1 (IF01), 4.1.1, or later. The older versions of GKLM are referred to as IBM Security Lifecycle Manager or SKLM in the documentation. The configuration information is the same for both GKLM and SKLM.

*Table 87. Firewall recommendations for GKLM*

| <b>Port number</b>                                                                                                                            | <b>Protocol</b> | <b>Service</b>                                                          | <b>Components</b>                                                                                               |
|-----------------------------------------------------------------------------------------------------------------------------------------------|-----------------|-------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| • 9083                                                                                                                                        | TCP             | WebSphere Application Server                                            | mmsklmconfig command for retrieving server certificate chain                                                    |
| • SKLM 2.6: 9080<br>• SKLM 2.7: 443<br>• SKLM 3.0: 443<br>• SKLM 3.0.1: 443<br>• SKLM 4.0: 9443<br>• GKLM 4.1.0.1: 9443<br>• GKLM 4.1.1: 9443 | TCP             | SKLM and GKLM REST admin interface                                      | mmsklmconfig utility for configuring IBM Storage Scale                                                          |
| • 5696                                                                                                                                        | TCP             | SKLM and GKLM Key Management Interoperability Protocol (KMIP) interface | IBM Storage Scale daemon for retrieving encryption keys, mmsklmconfig utility for configuring IBM Storage Scale |

## **Firewall recommendations for Thales Vormetric Data Security Manager (DSM)**

The file encryption feature in IBM Storage Scale uses two ports to communicate with the Thales Vormetric Data Security Manager (DSM) product.

DSM is one of the products that IBM Storage Scale supports as a Remote Key Management server for file encryption. The following table lists the recommended ports:

*Table 88. Firewall recommendations for DSM*

| <b>Port Number</b> | <b>Protocol</b> | <b>Service</b>                                                | <b>Components</b>                                                         |
|--------------------|-----------------|---------------------------------------------------------------|---------------------------------------------------------------------------|
| 8445               | TCP             | DSM administration web GUI                                    | The <b>mmsklmconfig</b> command for retrieving a server certificate chain |
| 5696               | TCP             | DSM Key Management Interoperability Protocol (KMIP) interface | The IBM Storage Scale daemon for retrieving encryption keys               |

For more information see [“Preparation for encryption” on page 742](#).

## Firewall recommendations for Performance Monitoring tool

The IBM Storage Scale system uses the following ports for the Performance Monitoring tool to work.

*Table 89. Recommended port numbers that can be used for Performance Monitoring tool*

| Port Number | Protocol    | Service Name                | Components that are involved in communication                                                                                                                           |
|-------------|-------------|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4739        | TCP and UDP | Performance Monitoring tool | Intra-cluster. This port needs to be accessible by all sensor nodes that are sending performance monitoring data.                                                       |
| 8123        | TCP         | Object Metric collection    | Intra-cluster                                                                                                                                                           |
| 8124        | TCP         | Object Metric collection    | Intra-cluster                                                                                                                                                           |
| 8125        | UDP         | Object Metric collection    | Intra-cluster                                                                                                                                                           |
| 8126        | TCP         | Object Metric collection    | Intra-cluster                                                                                                                                                           |
| 8127        | TCP         | Object Metric collection    | Intra-cluster                                                                                                                                                           |
| 9085        | TCP         | Performance Monitoring Tool | Intra-cluster. This port needs to be open to all nodes where the performance collector is running. This port is used for internal communication between the collectors. |
| 9980        | TCP         | Performance Monitoring Tool | Intra-cluster                                                                                                                                                           |

**Important:**

- The 4739 port needs to be open when a collector is installed.
- The 9085 port needs to be open when there are two or more collectors.
- If the port 9980 is closed, accessing the collector remotely, or connecting external tools or even connecting another instance of the GUI remotely, is not possible.

## Firewall considerations for Active File Management (AFM)

Active File Management (AFM) allows one or more IBM Storage Scale clusters, or a non-IBM Storage Scale NFS source, to exchange file data. File data exchange between clusters can accomplish many goals, one of which is to allow for disaster recovery.

For AFM data transfers, either NFS or NSD is used as the transport protocol.

- For port requirements of NFS, see [“Firewall recommendations for protocol access” on page 987](#).
- For port requirements of NSD, see [“Firewall recommendations for internal communication among nodes” on page 985](#).

## Firewall considerations for remote mounting of file systems

IBM Storage Scale clusters can access file systems on other IBM Storage Scale clusters using remote mounts.

Remote mounts can be used in the following ways.

- All nodes in the IBM Storage Scale cluster requiring access to another cluster's file system must have a physical connection to the disks containing file system data. This is typically done through a storage area network (SAN).
- All nodes in the IBM Storage Scale cluster requiring access to another cluster's file system must have a virtual connection through an NSD server.

In both cases, all nodes in the cluster requiring access to another cluster's file system must be able to open a TCP/IP connection to every node in the other cluster. For information on the basic GPFS cluster operation port requirements, see [“Firewall recommendations for internal communication among nodes” on page 985](#).

**Note:** Each cluster participating in a remote mount might reside on the same internal network or on a separate network from the host cluster. From a firewall standpoint, this means that the host cluster might need ports to be opened to a number of external networks, depending on how many separate clusters are accessing the host.

## Firewall recommendations for using IBM Storage Protect with IBM Storage Scale

The IBM Storage Scale **mmbackup** command is used to back up file systems and filesets to an externally located IBM Storage Protect server. IBM Storage Protect for Space Management is used with the IBM Storage Scale policy engine to migrate data to secondary storage pools residing on an IBM Storage Protect server.

Both functions require an open path for communication between the nodes designated for use with **mmbackup** or HSM policies and the external IBM Storage Protect server. The port requirement listed in the following table can be viewed in the `dsm.sys` configuration file also.

*Table 90. Required port number for mmbackup and IBM Storage Protect for Space Management connectivity to IBM Spectrum Protect server*

| Port number | Protocol | Service name | Components involved in communication                                |
|-------------|----------|--------------|---------------------------------------------------------------------|
| 1500        | TCP      | TSM          | IBM Storage Protect Backup-Archive client communication with server |

For information on port requirements specific to the server end, see IBM Storage Protect documentation.

## Firewall considerations for using IBM Spectrum Archive with IBM Storage Scale

The IBM Spectrum Archive software is installed on a node or a group of nodes in an IBM Storage Scale cluster.

This requires that each IBM Spectrum Archive node can communicate with the rest of the cluster using the ports required for basic GPFS cluster operations. For more information, see [“Firewall recommendations for internal communication among nodes” on page 985](#). In addition to this, IBM Spectrum Archive communicates with RPC. For RPC related port requirements, see [“Firewall recommendations for protocol access” on page 987](#).

IBM Spectrum Archive can connect to tape drives using a SAN or a direct connection.

## Firewall recommendations for call home

This topic describes port access and firewall protection during call home activities.

IBM Storage Scale IBM Support server is accessible through port 443. The following table lists the port and the Host/IP for communicating with the IBM Support server.

*Table 91. Recommended port numbers that can be used for call home*

| Port Number | Function  | Protocol    | Host/ IP                                                              |
|-------------|-----------|-------------|-----------------------------------------------------------------------|
| 443         | Call home | TCP and UDP | esupport.ibm.com:<br>• 192.148.6.11<br>• 2620:1f7:c010:1:1:1:<br>1:11 |

**Note:**

esupport.ibm.com works for the Call Home uploads as a proxy by forwarding data to ECuRep. Therefore, allowing any direct connections to ECuRep IPs is not required.

## Examples of how to open firewall ports

Use these examples as a reference for opening firewall ports on different operating systems. Restrict port traffic to only the needed network or adapters.

### Red Hat Enterprise Linux and CentOS

- Issue the following command to list currently open ports.

```
firewall-cmd --list-ports
```

- Issue the following command to list zones.

```
firewall-cmd --get-zones
```

- Issue the following command to list the zone that contains eth0.

```
firewall-cmd --get-zone-of-interface=eth0
```

- Issue the following command to open port 1191 for TCP traffic.

```
firewall-cmd --add-port 1191/tcp
```

- Issue the following command to open port 1191 for TCP traffic after restart. Use this command to make changes persistent.

```
firewall-cmd --permanent --add-port 1191/tcp
```

- Issue the following command to open a range a range of ports.

```
firewall-cmd --permanent --add-port 60000-61000/tcp
```

- Issue the following command to stop and start the firewall.

```
systemctl stop firewalld
```

```
systemctl start firewalld
```

### SLES

- Open the YaST tool by issuing the following command: **yast**

2. Click **Security and Users > Firewall**.
3. Select the **Allowed Services** tab and click **Advanced**.
4. Enter the wanted port range in the **from-port-start:to-port-end** format and specify the protocol (TCP or UDP). For example, enter **60000:60010** to open ports **60000 - 60010**.
5. Click **OK** to close the Advanced dialog box.
6. Click **Next** and review the summary of your changes.
7. Click **Finish** to apply your changes.

### **Ubuntu and Debian**

- Issue the following command to open port **1191** for TCP traffic.

```
sudo ufw allow 1191/tcp
```

- Issue the following command to open a range of ports.

```
sudo ufw allow 60000:61000/tcp
```

- Issue the following command to stop and start Uncomplicated Firewall (UFW).

```
sudo ufw disable
```

```
sudo ufw enable
```

### **Microsoft Windows 2008 R2**

1. Open the Windows Firewall utility: **Control Panel > Administrative Tools > Windows Firewall with Advanced Security**
2. Add new inbound and outbound rules as needed.

### **Firewall configuration by using iptables**

The **iptables** utility is available on most Linux distributions to set firewall rules and policies. These Linux distributions include Red Hat Enterprise Linux 6.8, Red Hat Enterprise Linux 7.x, CentOS 7.x, SLES 12, Ubuntu, and Debian. Before you use these commands, check which firewall zones might be enabled by default. Depending upon the zone setup, the INPUT and OUTPUT terms might need to be renamed to match a zone for the wanted rule. See the following Red Hat Enterprise Linux 7.x example for one such case.

- Issue the following command to list the current firewall policies.

```
sudo iptables -S
```

```
sudo iptables -L
```

- Issue the following command to open port **1191** (GPFS) for inbound TCP traffic from internal subnet **172.31.1.0/24**.

```
sudo iptables -A INPUT -p tcp -s 172.31.1.0/24 --dport 1191 -j ACCEPT
```

- Issue the following command to open port **1191** (GPFS) for outbound TCP traffic to internal subnet **172.31.1.0/24**.

```
sudo iptables -A OUTPUT -p tcp -d 172.31.1.0/24 --sport 1191 -j ACCEPT
```

- Issue the following command to open port **445** (SMB) for outbound TCP traffic to external subnet **10.11.1.0/24** and only for adapter **eth1**.

```
sudo iptables -A OUTPUT -o eth1 -p tcp -d 10.11.1.0/24 --sport 445 -j ACCEPT
```

- Issue the following command to open port 445 (SMB) for inbound TCP traffic to a range of CES IP addresses (10.11.1.5 through 10.11.1.11) and only for adapter eth1.

```
sudo iptables -A INPUT -i eth1 -p tcp -m iprange --dst-range 10.11.1.5-10.11.1.11 --dport 445 -j ACCEPT
```

- Issue the following command to allow an internal network, eth1, to communicate with an external network, eth0.

```
sudo iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
```

- [Red Hat Enterprise Linux 7.x specific] Issue the following command to open port 10080 for inbound traffic from subnet 10.18.0.0/24 on eth1 within the public zone.

```
iptables -A IN_public_allow -i eth1 -p tcp -s 10.18.0.0/24 --dport 10080 -j ACCEPT
```

- Issue the following command to save firewall rule changes to persist across a restart.

```
sudo iptables-save
```

- Issue the following command to stop and start Uncomplicated Firewall (UFW).

```
service iptables stop
```

```
service iptables start
```

For information on how CES IP addresses are aliased to network adapters, see [“CES IP aliasing to network adapters on protocol nodes” on page 53](#).

## Supported web browser versions and web browser settings for GUI

To access the management GUI, you must ensure that your web browser is supported and has the appropriate settings enabled.

The management GUI supports the following web browsers:

- Mozilla Firefox 79
- Mozilla Firefox Extended Support Release (ESR) 68
- Microsoft Edge 84
- Google Chrome 84

IBM supports higher versions of the browsers if the vendors do not remove or disable function that the product relies upon. For browser levels higher than the versions that are certified with the product, customer support accepts usage-related and defect-related service requests. If the support center cannot re-create the issue, support might request the client to re-create the problem on a certified browser version. Defects are not accepted for cosmetic differences between browsers or browser versions that do not affect the functional behavior of the product. If a problem is identified in the product, defects are accepted. If a problem is identified with the browser, IBM might investigate potential solutions or workarounds that the client can implement until a permanent solution becomes available.

**Note:** For the management GUI to function properly, JavaScript and cookies must be enabled in your web browser.

---

# Chapter 63. GUI limitations

The following are the limitations of the IBM Storage Scale GUI:

- The GUI supports all operating system versions that are supported by core GPFS (`gpfs.base`) except AIX and Windows.
- Up to 1000 nodes are supported per cluster.
- Up to three GUI nodes are supported per cluster.
- The GUI supports a subset of the CLI functionality. Additional capabilities will be added in the future releases of the product.
- The Object management panels do not support configurations with Keystone V2 API.
- One GUI instance supports a single cluster.
- In an IBM Storage Scale and Elastic Storage Server (ESS) mixed support environment, the ESS GUI must manage the whole cluster to display the ESS-specific pages in the GUI.
- The GUI is not supported on the cluster that runs on IBM Storage Scale release earlier than 4.2.0.0. The GUI supports IBM Storage Scale release 4.2.0.0 or later. Issue the `mmlsconfig` command to see the value that is set for the `minReleaseLevel` attribute. Use the `mmchconfig release=LATEST` command and restart the GUI to make the management GUI fully operational at the new code level. As changing the minimum release level affects the cluster behavior, refer the `mmchconfig` command man page and other related topics before you make this configuration change.
- The GUI node must be a homogeneous stack. That is, all packages must be of the same release. For example, do not mix the 5.0.4 GUI rpm with a 5.0.3 base rpm. However, GUI PTFs and interim fixes (efixes) can usually be applied without having to install the corresponding PTF or interim fix of the base package. This is helpful if you just want to get rid of a GUI issue without changing anything on the base layer.



# **Accessibility features for IBM Storage Scale**

---

Accessibility features help users who have a disability, such as restricted mobility or limited vision, to use information technology products successfully.

## **Accessibility features**

---

The following list includes the major accessibility features in IBM Storage Scale:

- Keyboard-only operation
- 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.

## **Keyboard navigation**

---

This product uses standard Microsoft Windows navigation keys.

## **IBM and accessibility**

---

See the [IBM Human Ability and Accessibility Center \(www.ibm.com/able\)](http://www.ibm.com/able) for more information about the commitment that IBM has to accessibility.



## 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\_.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

## 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).

Intel is a trademark of Intel Corporation or its subsidiaries in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

The registered trademark Linux is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Red Hat, OpenShift®, and Ansible are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of the Open Group in the United States and other countries.

## Terms and conditions for product documentation

---

Permissions for the use of these publications are granted subject to the following terms and conditions.

## **IBM Privacy Policy**

At IBM we recognize the importance of protecting your personal information and are committed to processing it responsibly and in compliance with applicable data protection laws in all countries in which IBM operates.

Visit the IBM Privacy Policy for additional information on this topic at <https://www.ibm.com/privacy/details/us/en/>.

## **Applicability**

These terms and conditions are in addition to any terms of use for the IBM website.

### **Personal use**

You can reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You cannot distribute, display, or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

### **Commercial use**

You can reproduce, distribute, and display these publications solely within your enterprise provided that all proprietary notices are preserved. You cannot 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 that are 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 cannot download, export, or reexport 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.



# Glossary

---

This glossary provides terms and definitions for IBM Storage Scale.

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.

For other terms and definitions, see the [IBM Terminology website](http://www.ibm.com/software/globalization/terminology) ([www.ibm.com/software/globalization/terminology](http://www.ibm.com/software/globalization/terminology)) (opens in new window).

## B

### **block utilization**

The measurement of the percentage of used subblocks per allocated blocks.

## C

### **cluster**

A loosely coupled collection of independent systems (nodes) organized into a network for the purpose of sharing resources and communicating with each other. See also *GPFS cluster*.

### **cluster configuration data**

The configuration data that is stored on the cluster configuration servers.

### **Cluster Export Services (CES) nodes**

A subset of nodes configured within a cluster to provide a solution for exporting GPFS file systems by using the Network File System (NFS), Server Message Block (SMB), and Object protocols.

### **cluster manager**

The node that monitors node status using disk leases, detects failures, drives recovery, and selects file system managers. The cluster manager must be a quorum node. The selection of the cluster manager node favors the quorum-manager node with the lowest node number among the nodes that are operating at that particular time.

**Note:** The cluster manager role is not moved to another node when a node with a lower node number becomes active.

### **clustered watch folder**

Provides a scalable and fault-tolerant method for file system activity within an IBM Storage Scale file system. A clustered watch folder can watch file system activity on a fileset, inode space, or an entire file system. Events are streamed to an external Kafka sink cluster in an easy-to-parse JSON format. For more information, see the *mmwatch command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

### **control data structures**

Data structures needed to manage file data and metadata cached in memory. Control data structures include hash tables and link pointers for finding cached data; lock states and tokens to implement distributed locking; and various flags and sequence numbers to keep track of updates to the cached data.

## D

### **Data Management Application Program Interface (DMAPI)**

The interface defined by the Open Group's XDSM standard as described in the publication *System Management: Data Storage Management (XDSM) API Common Application Environment (CAE) Specification C429*, The Open Group ISBN 1-85912-190-X.

**deadman switch timer**

A kernel timer that works on a node that has lost its disk lease and has outstanding I/O requests. This timer ensures that the node cannot complete the outstanding I/O requests (which would risk causing file system corruption), by causing a panic in the kernel.

**dependent fileset**

A fileset that shares the inode space of an existing independent fileset.

**disk descriptor**

A definition of the type of data that the disk contains and the failure group to which this disk belongs. See also *failure group*.

**disk leasing**

A method for controlling access to storage devices from multiple host systems. Any host that wants to access a storage device configured to use disk leasing registers for a lease; in the event of a perceived failure, a host system can deny access, preventing I/O operations with the storage device until the preempted system has reregistered.

**disposition**

The session to which a data management event is delivered. An individual disposition is set for each type of event from each file system.

**domain**

A logical grouping of resources in a network for the purpose of common management and administration.

**E****ECKD**

See *extended count key data (ECKD)*.

**ECKD device**

See *extended count key data device (ECKD device)*.

**encryption key**

A mathematical value that allows components to verify that they are in communication with the expected server. Encryption keys are based on a public or private key pair that is created during the installation process. See also *file encryption key*, *master encryption key*.

**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.

**extended count key data device (ECKD device)**

A disk storage device that has a data transfer rate faster than some processors can utilize and that is connected to the processor through use of a speed matching buffer. A specialized channel program is needed to communicate with such a device. See also *fixed-block architecture disk device*.

**F****failback**

Cluster recovery from failover following repair. See also *failover*.

**failover**

(1) The assumption of file system duties by another node when a node fails. (2) The process of transferring all control of the ESS to a single cluster in the ESS when the other clusters in the ESS fails. See also *cluster*. (3) The routing of all transactions to a second controller when the first controller fails. See also *cluster*.

**failure group**

A collection of disks that share common access paths or adapter connections, and could all become unavailable through a single hardware failure.

**FEK**

See *file encryption key*.

**fileset**

A hierarchical grouping of files managed as a unit for balancing workload across a cluster. See also *dependent fileset*, *independent fileset*.

**fileset snapshot**

A snapshot of an independent fileset plus all dependent filesets.

**file audit logging**

Provides the ability to monitor user activity of IBM Storage Scale file systems and store events related to the user activity in a security-enhanced fileset. Events are stored in an easy-to-parse JSON format. For more information, see the *mmaudit command* in the *IBM Storage Scale: Command and Programming Reference Guide*.

**file clone**

A writable snapshot of an individual file.

**file encryption key (FEK)**

A key used to encrypt sectors of an individual file. See also *encryption key*.

**file-management policy**

A set of rules defined in a policy file that GPFS uses to manage file migration and file deletion. See also *policy*.

**file-placement policy**

A set of rules defined in a policy file that GPFS uses to manage the initial placement of a newly created file. See also *policy*.

**file system descriptor**

A data structure containing key information about a file system. This information includes the disks assigned to the file system (*stripe group*), the current state of the file system, and pointers to key files such as quota files and log files.

**file system descriptor quorum**

The number of disks needed in order to write the file system descriptor correctly.

**file system manager**

The provider of services for all the nodes using a single file system. A file system manager processes changes to the state or description of the file system, controls the regions of disks that are allocated to each node, and controls token management and quota management.

**fixed-block architecture disk device (FBA disk device)**

A disk device that stores data in blocks of fixed size. These blocks are addressed by block number relative to the beginning of the file. See also *extended count key data device*.

**fragment**

The space allocated for an amount of data too small to require a full block. A fragment consists of one or more subblocks.

**G****GPUDirect Storage**

IBM Storage Scale's support for NVIDIA's GPUDirect Storage (GDS) enables a direct path between GPU memory and storage. File system storage is directly connected to the GPU buffers to reduce latency and load on CPU. Data is read directly from an NSD server's pagepool and it is sent to the GPU buffer of the IBM Storage Scale clients by using RDMA.

**global snapshot**

A snapshot of an entire GPFS file system.

**GPFS cluster**

A cluster of nodes defined as being available for use by GPFS file systems.

**GPFS portability layer**

The interface module that each installation must build for its specific hardware platform and Linux distribution.

## **GPFS recovery log**

A file that contains a record of metadata activity and exists for each node of a cluster. In the event of a node failure, the recovery log for the failed node is replayed, restoring the file system to a consistent state and allowing other nodes to continue working.

## **I**

### **ill-placed file**

A file assigned to one storage pool but having some or all of its data in a different storage pool.

### **ill-replicated file**

A file with contents that are not correctly replicated according to the desired setting for that file. This situation occurs in the interval between a change in the file's replication settings or suspending one of its disks, and the restripe of the file.

### **independent fileset**

A fileset that has its own inode space.

### **indirect block**

A block containing pointers to other blocks.

### **inode**

The internal structure that describes the individual files in the file system. There is one inode for each file.

### **inode space**

A collection of inode number ranges reserved for an independent fileset, which enables more efficient per-fileset functions.

### **ISKLM**

IBM Security Key Lifecycle Manager. For GPFS encryption, the ISKLM is used as an RKM server to store MEKs.

## **J**

### **journalized file system (JFS)**

A technology designed for high-throughput server environments, which are important for running intranet and other high-performance e-business file servers.

### **junction**

A special directory entry that connects a name in a directory of one fileset to the root directory of another fileset.

## **K**

### **kernel**

The part of an operating system that contains programs for such tasks as input/output, management and control of hardware, and the scheduling of user tasks.

## **M**

### **master encryption key (MEK)**

A key used to encrypt other keys. See also *encryption key*.

### **MEK**

See *master encryption key*.

### **metadata**

Data structures that contain information that is needed to access file data. Metadata includes inodes, indirect blocks, and directories. Metadata is not accessible to user applications.

### **metanode**

The one node per open file that is responsible for maintaining file metadata integrity. In most cases, the node that has had the file open for the longest period of continuous time is the metanode.

**mirroring**

The process of writing the same data to multiple disks at the same time. The mirroring of data protects it against data loss within the database or within the recovery log.

**Microsoft Management Console (MMC)**

A Windows tool that can be used to do basic configuration tasks on an SMB server. These tasks include administrative tasks such as listing or closing the connected users and open files, and creating and manipulating SMB shares.

**multi-tailed**

A disk connected to multiple nodes.

**N****namespace**

Space reserved by a file system to contain the names of its objects.

**Network File System (NFS)**

A protocol, developed by Sun Microsystems, Incorporated, that allows any host in a network to gain access to another host or netgroup and their file directories.

**Network Shared Disk (NSD)**

A component for cluster-wide disk naming and access.

**NSD volume ID**

A unique 16-digit hex number that is used to identify and access all NSDs.

**node**

An individual operating-system image within a cluster. Depending on the way in which the computer system is partitioned, it may contain one or more nodes.

**node descriptor**

A definition that indicates how GPFS uses a node. Possible functions include: manager node, client node, quorum node, and nonquorum node.

**node number**

A number that is generated and maintained by GPFS as the cluster is created, and as nodes are added to or deleted from the cluster.

**node quorum**

The minimum number of nodes that must be running in order for the daemon to start.

**node quorum with tiebreaker disks**

A form of quorum that allows GPFS to run with as little as one quorum node available, as long as there is access to a majority of the quorum disks.

**non-quorum node**

A node in a cluster that is not counted for the purposes of quorum determination.

**Non-Volatile Memory Express (NVMe)**

An interface specification that allows host software to communicate with non-volatile memory storage media.

**P****policy**

A list of file-placement, service-class, and encryption rules that define characteristics and placement of files. Several policies can be defined within the configuration, but only one policy set is active at one time.

**policy rule**

A programming statement within a policy that defines a specific action to be performed.

**pool**

A group of resources with similar characteristics and attributes.

**portability**

The ability of a programming language to compile successfully on different operating systems without requiring changes to the source code.

**primary GPFS cluster configuration server**

In a GPFS cluster, the node chosen to maintain the GPFS cluster configuration data.

**private IP address**

An IP address used to communicate on a private network.

**public IP address**

An IP address used to communicate on a public network.

**Q****quorum node**

A node in the cluster that is counted to determine whether a quorum exists.

**quota**

The amount of disk space and number of inodes assigned as upper limits for a specified user, group of users, or fileset.

**quota management**

The allocation of disk blocks to the other nodes writing to the file system, and comparison of the allocated space to quota limits at regular intervals.

**R****Redundant Array of Independent Disks (RAID)**

A collection of two or more disk physical drives that present to the host an image of one or more logical disk drives. In the event of a single physical device failure, the data can be read or regenerated from the other disk drives in the array due to data redundancy.

**recovery**

The process of restoring access to file system data when a failure has occurred. Recovery can involve reconstructing data or providing alternative routing through a different server.

**remote key management server (RKM server)**

A server that is used to store master encryption keys.

**replication**

The process of maintaining a defined set of data in more than one location. Replication consists of copying designated changes for one location (a source) to another (a target) and synchronizing the data in both locations.

**RKM server**

See *remote key management server*.

**rule**

A list of conditions and actions that are triggered when certain conditions are met. Conditions include attributes about an object (file name, type or extension, dates, owner, and groups), the requesting client, and the container name associated with the object.

**S****SAN-attached**

Disk that are physically attached to all nodes in the cluster using Serial Storage Architecture (SSA) connections or using Fibre Channel switches.

**Scale Out Backup and Restore (SOBAR)**

A specialized mechanism for data protection against disaster only for GPFS file systems that are managed by IBM Storage Protect for Space Management.

**secondary GPFS cluster configuration server**

In a GPFS cluster, the node chosen to maintain the GPFS cluster configuration data in the event that the primary GPFS cluster configuration server fails or becomes unavailable.

**Secure Hash Algorithm digest (SHA digest)**

A character string used to identify a GPFS security key.

**session failure**

The loss of all resources of a data management session due to the failure of the daemon on the session node.

**session node**

The node on which a data management session was created.

**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.

**snapshot**

An exact copy of changed data in the active files and directories of a file system or fileset at a single point in time. See also *fileset snapshot*, *global snapshot*.

**source node**

The node on which a data management event is generated.

**stand-alone client**

The node in a one-node cluster.

**storage area network (SAN)**

A dedicated storage network tailored to a specific environment, combining servers, storage products, networking products, software, and services.

**storage pool**

A grouping of storage space consisting of volumes, logical unit numbers (LUNs), or addresses that share a common set of administrative characteristics.

**stripe group**

The set of disks comprising the storage assigned to a file system.

**striping**

A storage process in which information is split into blocks (a fixed amount of data) and the blocks are written to (or read from) a series of disks in parallel.

**subblock**

The smallest unit of data accessible in an I/O operation, equal to one thirty-second of a data block.

**system storage pool**

A storage pool containing file system control structures, reserved files, directories, symbolic links, special devices, as well as the metadata associated with regular files, including indirect blocks and extended attributes. The system storage pool can also contain user data.

**T****token management**

A system for controlling file access in which each application performing a read or write operation is granted some form of access to a specific block of file data. Token management provides data consistency and controls conflicts. Token management has two components: the token management server, and the token management function.

**token management function**

A component of token management that requests tokens from the token management server. The token management function is located on each cluster node.

**token management server**

A component of token management that controls tokens relating to the operation of the file system. The token management server is located at the file system manager node.

**transparent cloud tiering (TCT)**

A separately installable add-on feature of IBM Storage Scale that provides a native cloud storage tier. It allows data center administrators to free up on-premise storage capacity, by moving out cooler data to the cloud storage, thereby reducing capital and operational expenditures.

**twin-tailed**

A disk connected to two nodes.

**U****user storage pool**

A storage pool containing the blocks of data that make up user files.

**V****VFS**

See *virtual file system*.

**virtual file system (VFS)**

A remote file system that has been mounted so that it is accessible to the local user.

**virtual node (vnode)**

The structure that contains information about a file system object in a virtual file system (VFS).

**W****watch folder API**

Provides a programming interface where a custom C program can be written that incorporates the ability to monitor inode spaces, filesets, or directories for specific user activity-related events within IBM Storage Scale file systems. For more information, a sample program is provided in the following directory on IBM Storage Scale nodes: /usr/lpp/mmfs/samples/util called tswf that can be modified according to the user's needs.

# Index

## Special Characters

/access/control [515](#)  
/etc/group [507](#)  
/etc/passwd [507](#)  
/var/mmfs/ssl/id\_rsa.pub [513](#), [521](#)

## A

access  
    control [515](#)  
access ACL [462](#)  
access control list [489](#)  
access control lists  
    administering [465](#)  
    allow type [466](#)  
    applying [471](#)  
    authorize file protocol users [475](#)  
    authorizing object users [489](#), [495](#)  
    authorizing protocol users  
        limitations [496](#)  
    best practices [480](#)  
    change NFS V4 ACL [472](#)  
    changing [465](#)  
    DELETE [468](#)  
    DELETE\_CHILD [468](#)  
    deleting [465](#), [472](#)  
    deny type [466](#)  
    display NFS V4 ACL [471](#)  
    displaying [463](#)  
    exceptions [472](#)  
    export-level ACLs [476](#)  
inheritance  
    DirInherit [467](#)  
    FileInherit [467](#)  
    Inherited [467](#)  
    InheritOnly [467](#)  
inheritance flags [479](#)  
limitations [472](#)  
Linux [474](#)  
managing [461](#)  
NFS V4 [461](#), [465](#)  
NFS V4 syntax [466](#)  
object ACLs  
    creating write ACLs [495](#)  
required permissions [482](#)  
setting [462](#), [464](#), [470](#)  
special names [467](#)  
traditional [461](#)  
translation [469](#)  
work with ACLs [486](#)  
access to file systems  
    access patterns of applications [69](#)  
accessibility features for IBM Storage Scale [1001](#)  
acl  
    managing quotas [489](#)  
ACTION [535](#)

activating quota limit checking [440](#)  
active commands  
    listing [203](#)  
Active Directory  
    authentication for file access [304](#)  
    authentication for overlapping unixmap ranges [312](#)  
active file management  
    FPO pool file placement [687](#)  
Active File management [159](#)  
Active File Management [962](#)  
active-active cluster  
    failback [637](#)  
    failover [637](#)  
    IBM TotalStorage  
        configuration [635](#)  
active-passive cluster  
    failback [641](#)  
    failover [641](#)  
    IBM TotalStorage  
        configuration [639](#)  
AD for file  
    prerequisites [306](#)  
AD for file authentication  
    AD with automatic ID mapping [307](#)  
    AD with RFC2307 ID mapping [308](#)  
AD-based authentication [314](#)  
adding  
    disks [276](#)  
adding nodes to a GPFS cluster [4](#)  
administering  
    GPFS file system [199](#), [200](#)  
administering files  
    using transparent cloud tiering [877](#)  
administration security [66](#)  
administration tasks [199](#)–[202](#)  
adminMode  
    requirements for administering GPFS [200](#)  
advanced administration [977](#)  
AFM  
    Administering [923](#)  
    AFM relationship  
        GPFS protocol [929](#)  
    cache cluster  
        gateway node [145](#)  
    configuration parameters [133](#)  
    creating  
        AFM relationship [928](#)  
        Downloading objects [956](#)  
        firewall considerations [994](#)  
        FPO pool file placement [687](#)  
        home cluster  
            NFS server [145](#)  
        NFS protocol  
            creating an AFM relationship [923](#)  
        Outband method [956](#)  
        setting up  
            cache cluster [929](#)

AFM (*continued*)  
     setting up (*continued*)  
         home cluster 928  
     setting up the cache cluster 925  
     setting up the home cluster 923

AFM DR  
     Administering 933  
     Changing NFS server at secondary 149  
     Converting  
         GPFS filesets to AFM DR 935  
     Converting AFM relationship to AFM DR 936  
     creating  
         AFM-DR relationship 933

AFM relationship  
     NFS protocol 925

AFM to cloud object storage  
     downloading objects 954  
     uploading objects 952

AFM to cloud object storage download statistics 962

AFM to cloud object storage upload statistics 962

allow access 515

Amazon Simple Storage Service IBM Storage Scale  
     cloudkit administering 921

appendOnly  
     directories 594  
     effects 594  
     files 594  
     integrated archive manager (IAM) modes 594

application programs  
     access patterns 69

applications for highly-available write cache 971

apply data placement policy 691

asynchronous mirroring  
     IBM ESS FlashCopy 642

attributes  
     adminMode 200  
     filesets  
         changing 592  
         useNSDserver 284

audit types 911

authentication  
     authentication for file access  
         AD with automatic ID mapping 307  
         AD with RFC2307 ID mapping 308  
         LDAP-based authentication 315  
         NIS-based authentication 322  
         set up ID map range 305  
     authentication for object access 293  
     configure file user authentication 323  
     deleting 333  
     limitations 335  
     listing 330  
     modifying 332  
     protocol user authentication  
         set up authentication servers 293  
     set up authentication servers  
         integrating with AD server 294  
         integrating with LDAP server 295  
     User-defined method of authentication 326  
     verifying 331

authentication considerations  
     NFSv4 based access 301

authentication limitations 335

authorizing protocol users  
     authorizing protocol users (*continued*)  
         authorize file protocol users 475  
         authorizing object users 489, 495  
         export-level ACLs 476  
         limitations 496  
         object ACLs  
             creating read ACLs 493  
             creating write ACLs 495  
             work with ACLs 486

auto recovery 731

auto-generated ID mappings  
     Windows 665

automating the maintenance activities 100

automount 217

autorecovery  
     QoS support 734

availability  
     disk 280

available write cache, highly- 971

**B**

back ends, RKM 742

Back-up option  
     Cloud services configuration 99

Backing up  
     Cloud data sharing database 99  
     Cloud services configuration 99

backing up a file system  
     tuning with mmbackup 247  
     using the GPFS policy engine 249  
     using the mmbackup command 244

backing up a fileset  
     using the mmbackup command 245

backing up file system configuration information  
     using the mmbackupconfig command 250

backup  
     file system  
         SOBAR 617  
         storage pools 576

backup applications  
     writing 250

Backup option  
     Cloud data sharing database 99

backup/restore and encryption 862

best practices  
     configuring AD with RFC2307 as the authentication method 314

bind user requirements 296

built-in functions  
     policy rules 546  
     types  
         date and time 551  
         extended attributes 546  
         numerical 551  
         string 549

**C**

cache  
     GPFS token system's effect on 68  
     GPFS usage 66  
     local read-only 975

cache (*continued*)  
    pageable memory for file attributes not in file cache [67](#)  
    pagepool [66](#)  
    total number of different file cached at one time [67](#)

cache cluster  
    gateway node [145](#)

cache purging, encryption key [858](#)

cache, highly available write [971](#)

call home  
    configuring [173](#), [174](#)  
    firewall [996](#)  
    firewall recommendations [996](#)  
    use case [176](#)

CCR (Clustered Configuration Repository)  
    failback with temporary loss [632](#)

certificate expiration  
    log messages [832](#)  
    Warnings [832](#)

Certificates  
    renew [835](#), [845](#)  
    server [838](#)

Certificates (Expiration error messages [835](#))

CES  
    configuration  
        file systems [60](#)  
        filesets [60](#)  
        nodes [51](#)  
        protocol service IP addresses [52](#)  
        shared root file system [49](#)  
        verification [59](#)  
    multiprotocol exports [356](#)  
    NFS export configuration  
        change [353](#)  
        changing [352](#)  
        create [351](#)  
    NFS exports  
        removal [353](#)  
    protocol services  
        disabling [291](#)  
    SMB configuration  
        export ACL [341](#)  
        exports [339](#)  
    SMB export configuration  
        changing [340](#)  
    SMB exports  
        removal [341](#)  
    SMB, S3, and NFS protocol services  
        starting [287](#)  
    start [652](#)  
    stop [652](#)  
    Swift Object protocol services  
        starting [290](#)

CES (Cluster Export Service) clusters  
    migration from CNFS [662](#)

CES (Cluster Export Services)  
    address distribution [654](#)  
    failover [654](#)  
    IP addresses [652](#)  
    management [651](#)  
    network configuration [652](#)  
    protocols  
        disable [655](#)  
        enable [655](#)  
    resume [655](#)

CES (Cluster Export Services) (*continued*)  
    setup [651](#)  
    shared root directory [651](#)  
    suspend [655](#)

CES (Cluster Export Services) implementing [651](#)

CES data disaster recovery  
    failback steps [680](#)  
    failover steps [680](#)

CES groups  
    add to cluster [61](#)

CES IP aliasing [53](#)

CES node  
    remove from cluster [61](#)

CES packages  
    deploying [58](#)

CES(Cluster Export Services)  
    HDFS protocol [662](#)  
    NFS protocol [656](#)  
    OBJ protocol [659](#)  
    SMB protocol [658](#)

change password [455](#)

changing  
    disk states [281](#)  
    hostnames [977](#)  
    IP addresses [977](#)  
    node names [977](#)  
    node numbers [977](#)  
    quotas [431](#)  
    replication [229](#), [230](#)

Changing gateway nodes in primary [149](#)

Changing NFS server at secondary [149](#)

changing quota limit checking [443](#)

Channel Bonding [69](#)

CHAR [550](#)

check  
    data locality [722](#)

checking  
    file systems [223](#)  
    quotas [436](#)

checking the  
    Cloud services database integrity [888](#)

checking the Cloud services DB  
    power outage [888](#)

child fileset [582](#)

chmod [462](#)

clauses  
    ACTION [535](#)  
    COMPRESS [535](#)  
    DIRECTORIES\_PLUS [536](#)  
    EXCLUDE [537](#)  
    FOR FILESET [537](#)  
    FROM POOL [537](#)  
    GROUP POOL [537](#)  
    LIMIT [538](#)  
    REPLICATE [538](#)  
    SET POOL [538](#)  
    SHOW [539](#)  
    THRESHOLD [539](#)  
    TO POOL [540](#)  
    WEIGHT [540](#)  
    WHEN [540](#)  
    WHERE [540](#)

clean cluster shutdown [36](#)

clean up files from cloud storage tier [883](#)

clones  
    file clones [613](#)  
Cloud data sharing database  
    backup option [99](#)  
cloud data sharing service  
    managing [906](#)  
cloud data sharing using  
    Transparent Cloud Tiering [902](#)  
Cloud Object Storage [159](#), [962](#)  
cloud object storage account  
    configuring [90](#)  
Cloud services  
    configure maintenance windows [100](#)  
    configuring and tuning [87](#)  
    define [94](#)  
    maintenance activities [100](#)  
    restore procedure [894](#)  
Cloud services (SOBAR) [889](#)  
Cloud services configuration  
    back-up option [99](#)  
    restore option [887](#)  
Cloud services configuration for multi-clouds and multi file systems [94](#)  
Cloud services database integrity [888](#)  
cloud storage tier  
    creating [90](#)  
    recall files [882](#)  
cloudkit  
    administering [917](#)  
    edit  
        cloud cluster [918](#)  
    JumpHost  
        GUI access [919](#)  
    manage [917](#)  
    managing [917](#)  
    mounting  
        GPFS remote mount [917](#)  
    scale out  
        cloud cluster [918](#)  
    unmounting  
        GPFS remote mount [918](#)  
cluster  
    disaster recovery [623](#)  
    minimum release level [24](#)  
Cluster Export Service (CES) clusters  
    migration from CNFS [662](#)  
Cluster Export Services (CES)  
    HDFS protocol [662](#)  
    NFS protocol [656](#)  
    OBJ protocol [659](#)  
    resume [655](#)  
    SMB protocol [658](#)  
    suspend [655](#)  
Cluster Export Services (CES))  
    address distribution [654](#)  
    failover [654](#)  
    IP addresses [652](#)  
    management [651](#)  
    network configuration [652](#)  
    protocols  
        disable [655](#)  
        enable [655](#)  
    setup [651](#)  
Cluster Export Services (CES)implementing [651](#)  
cluster quorum with tiebreaker [31](#)  
Clustered Configuration Repository (CCR)  
    failback with temporary loss [632](#)  
Clustered NFS (CNFS) environment  
    administration [648](#)  
    configuration [648](#)  
    failover [645](#)  
    implementing  
        Linux [645](#)  
    load balancing [646](#)  
    locking [646](#)  
    monitoring [645](#)  
    network setup [646](#)  
    setup [647](#)  
Clustered NFS environment (CNFS)  
    migration to CES [662](#)  
clustered NFS subsystem  
    using [503](#)  
clustered watch [129](#)  
clustered watch folder [129](#), [130](#)  
clusters  
    accessing file systems [505](#)  
    configuring [688](#)  
    exporting data [707](#)  
    exporting output data [707](#)  
    ingesting data [707](#)  
CNFS [503](#)  
CNFS (Cluster NFS environment  
    migration to CES [662](#)  
CNFS (Clustered NFS) environment  
    administration [648](#)  
    configuration [648](#)  
    failover [645](#)  
    implementing  
        Linux [645](#)  
    load balancing [646](#)  
    locking [646](#)  
    monitoring [645](#)  
    network setup [646](#)  
    setup [647](#)  
co-resident state migration [880](#)  
collecting  
    performance metrics [106](#)  
commands  
    active [203](#)  
    chmod [462](#)  
    cloudkit  
        edit filesystem [918](#)  
        grant filesystem [917](#)  
        grant guiAccess [919](#)  
        grant repository [920](#), [921](#)  
        revoke filesystem [918](#)  
    localityCopy [727](#)  
    mmaddcallback [531](#), [567](#)  
    mmadddisk [276](#), [527](#)  
    mmaddnode [4](#), [979](#)  
    mmapplypolicy [244](#), [528](#), [531](#), [553](#), [555](#), [557](#), [562](#), [567](#), [568](#), [573](#), [575](#), [605](#)  
    mmauth [22](#), [505](#), [508](#), [512](#), [513](#), [516](#), [518](#), [520](#), [521](#)  
    mmbackup [244](#), [245](#), [247](#), [248](#), [586](#)  
    mmbackupconfig [244](#)  
    mmces [652](#), [654](#), [655](#)  
    mmchattr [229](#)–[231](#), [240](#), [528](#)  
    mmchcluster [8](#), [978](#)

commands (*continued*)  
mmchconfig 21, 22, 31, 66, 501, 502, 505, 512, 518, 523, 651, 981  
mmchdisk 225, 240, 280, 281, 527  
mmcheckquota 425, 436, 441, 443  
mmchfileset 592  
mmchfs 229, 284, 425, 440, 443  
mmchnsd 283, 978  
mmchpolicy 531, 563–567, 736  
mmchqos 237  
mmcrcluster 1, 505  
mmcrfileset 588  
mmcrfs 217, 425, 440, 465, 527  
mmcrnsd 275, 276  
mmcrsnapshot 250, 601  
mmdefragfs 242, 243  
mmdelacl 465, 472  
mmddeldisk 276, 527, 528  
mmddelfileset 590  
mmdeflfs 222  
mmdeinode 5, 979  
mmdelesnapshot 607  
mmdf 241, 276, 523, 526, 529  
mmeditacl 465, 469, 470  
mmmedquota 425, 431  
mmexportfs 981  
mmfsck 223, 225, 276, 523  
mmgetacl 462–464, 469–472  
mmgetlocation 724  
mmimportfs 981  
mmlinkfileset 582, 588, 591, 592  
mmlsattr 229, 230, 528, 582, 593  
mmlscluster 2, 514  
mmlsconfig 523  
mmlsdisk 225, 280, 523, 981  
mmlsfileset 585, 586, 590, 593  
mmlsfs 228, 279, 440, 441, 523, 528  
mmlsmgr 33  
mmlsmount 223, 523  
mmlsnsd 275, 978  
mmmlspolicy 566  
mmmlsqos 237  
mmmlsquota 438  
mmmount 217, 218, 284, 514  
mmnetverify 215  
mmputacl 462, 464, 465, 470, 472  
mmquotaoff 440, 441  
mmquotaon 440  
mmremotecluster 505, 514, 521, 523  
mmremotesfs 284, 505, 514, 523  
mmrepquota 444  
mmrestorefs 586  
mmrestripefile 528  
mmrestripefs  
completion time 204  
mmrpldisk 279, 527  
mmsetquota 425  
mmshutdown 35, 651, 977, 978  
mmssnapdir 586, 601  
mmstartup 35, 651, 978  
mmumount 221  
mmunlinkfileset 582, 590, 592  
mmuserauth 293, 300, 303  
common GPFS command principles 201, 202  
communications I/O  
Linux nodes 71  
COMPRESS 535  
CONCAT 550  
configuration  
ID mapping 333  
configuration and tuning settings  
access patterns 69  
aggregate network interfaces 69  
AIX settings  
use with Oracle 72  
clock synchronization 65  
communications I/O 71  
disk I/O 72  
general settings 65  
GPFS helper threads 71  
GPFS I/O 72  
GPFS page pool 66  
Jumbo Frames 71  
Linux settings  
communications I/O 71  
disk I/O 72  
GPFS helper threads 71  
memory considerations 71  
updatedb considerations 71  
monitoring GPFS I/O performance  
65  
security 66  
swap space 70  
TCP window 71  
use with Oracle 72  
configuration tasks  
CES 49, 60  
CES nodes 51  
CES protocol  
service IP addresses 52  
CES shared root file system 49  
CES verification 59  
changing NFS exports 352  
changing SMB exports 340  
disabling protocol services 291  
file systems 60  
filesets 60  
NFS export removal 353  
setting quotas 433  
SMB and NFS protocols 356  
SMB export ACL creation 341  
SMB export creation 339  
SMB export removal 341  
configure authentication  
set identity management modes 398  
configure GUI authentication  
multi-factor authentication 458  
Configure ISV  
multi-factor authentication 459  
configured services 331  
configuring  
cloud data sharing node 88  
cloud object storage 90  
Cloud services 95  
Cloud services node 88  
clustered watch folder 129  
immutable fileset 111  
transparent cloud tiering 87

configuring (*continued*)  
    transparent cloud tiering node 88

Configuring  
    SUDO 28  
        with LDAP ID mapping 314

Configuring a key manager  
    Cloud services 95

configuring a maintenance window 100

configuring AD with RFC2307 314

configuring AD with RFC2307 as the authentication method  
    best practices 314

configuring and tuning  
    cloud data sharing 87  
        transparent cloud tiering 87, 877

configuring call home 173

configuring call home automatically 174

configuring call home manually 174

configuring certificate-based authentication and locked vaults 116

configuring Cloud services 96

Configuring for WORM solutions 112

configuring GPFS clusters 688

Configuring ID mappings in Active Directory Users and Computers 666

configuring ID mappings in IDMU  
    Windows 670

configuring Kerberos based NFS access  
    prerequisites 302

configuring locked vaults and certificate-based authentication 116

configuring protocols  
    IBM Storage Scale 509

configuring protocols on a remote cluster 509

configuring sudo 28

Configuring Transparent cloud tiering on a remotely mounted client 109

configuring WORM solutions 116

considerations for changing  
    range size 306  
    the ID map range 306

consistency groups  
    IBM Storage Scale 624

container pairs for Transparent cloud tiering 96

control file permission 462

Converting  
    GPFS filesets to AFM DR 935

Converting AFM relationship to AFM DR 936

create data placement policy 691

create IBM Storage Scale file system and pools 690

create users 453

creating  
    AFM relationship 928  
        AFM-based DR relationship 933  
    data and metadata vaults 115  
        file clones 613  
        immutable fileset 111  
        quota reports 444  
        snapshots 601

Creating  
    AFM to cloud object storage relationship 940

creating a container set 96

Creating Cloud services 94

creating cloud storage access points  
    Transparent cloud tiering 92

creating CSAPs  
    Transparent cloud tiering 92

creating data and metadata containers 96

creating locked vaults  
    configuring WORM 115

CURRENT\_DATE 551

CURRENT\_TIMESTAMP 540, 551

**D**

data  
    multiple versions 624

data deletion, secure 735

data integrity  
    IBM Storage Scale 624

data locality 722

data locality restoration 728

data locality restore 721

data placement policy 691

data protection 735

data recovery 888

data replication  
    changing 230

data security limitations  
    data, security limitations 875

database recovery  
    transparent cloud tiering 888

Database workloads  
    configuration 706  
    tuning 706

DAY 551

DAYOFWEEK 551

DAYOFYEAR 551

DAYS 551

DAYSMONTH 551

DAYSYEAR 551

deactivating quota limit checking 441

declustered array stanza 202

default ACL 462

default quotas 427

DELETE rule 531, 539

deleting  
    a GPFS cluster 5  
    file systems 222  
    nodes from a GPFS cluster 5  
    snapshots 607

deleting a cloud storage account 90

deleting a CSAP  
    configuring Transparent cloud tiering 92

deleting cloud objects 884

deleting files  
    manually 884

deletion of data, secure 735

deny access 515

dependent fileset 515

deploy WORM on IBM Storage Scale 110

deploying  
    WORM solutions 117

deploying WORM solutions 116

Deploying WORM solutions  
    IBM Storage Scale 112  
        set up private key and private certificate 113

designating  
    transparent cloud tiering node 88

detect system changes  
use case 179

Direct I/O caching policy 231

DIRECTORIES\_PLUS 536

directory server 297

DirInherit 467

disable  
file audit logging 127

disabling  
clustered watch 129, 130

GUI access 919

Persistent Reserve 284

QOS 397

repository access 920

disaster recovery  
establishing 624

GPFS replication  
configuring 627

IBM ESS FlashCopy 642

IBM TotalStorage  
active-active cluster 635

active-passive cluster 639

overview 623

disaster recovery procedure (Cloud services 889)

disk availability 280

disk descriptor 527

disk descriptors 276, 278

disk discovery 284

disk failure  
stopping auto recovery 716

disk failures 716

Disk offline  
Writing data 279

disk replacement 729

disk state  
changing 281

displaying 280

disk status 280

diskFailure Event 732

diskIOErr  
mmaddcallback 284

disks  
adding 276

availability 280

deleting 276

displaying information 275

ENOSPC 279

failure 240

fragmentation 242

I/O settings 72

managing 275

maximum number 275

replacing 278, 279

status 280

storage pool assignment  
changing 527

strict replication 279

displaying  
access control lists 463

disk fragmentation 242

disk states 280

disks 275

quotas 438

distributedTokenServer 981

dynamic validation of descriptors on disk 225

## E

editing  
cloud cluster 918

EINVAL 532

enabling  
AFM caching 921

clustered watch 129

filesets 126

GUI access 919

Persistent Reserve 284

QOS 397

repository access 920

enabling cluster for IPv6  
IPv6, enabling a cluster for 980

encrypted file  
remote access 774, 778, 807

encryption  
encryption-enabled environment 747

key rotation 858

local read-only cache (LROC) 862

regular setup 788

secure deletion 858

simplified setup 747, 757

simplified tasks 780

standards compliance 860

Encryption  
external pools 863

Encryption (IBM Spectrum Archive Enterprise Edition 863)

Encryption (IBM Spectrum Protect 863)

Encryption (IBM Spectrum Scale Transparent Cloud Tiering 863)

encryption and backup/restore 862

encryption and FIPS compliance 860

encryption and NIST compliance 861

encryption and secure deletion 856

encryption and snapshots 862

encryption hints 856

ENCRYPTION IS policy rule 736

encryption key cache purging 858

encryption keys 735, 858

Encryption limitations 863

encryption policies 736

encryption policies, rewrapping 741

encryption policy example 740

encryption policy rules 736

Encryption requirements 863

encryption setup requirements 742

encryption-enabled environment  
regular setup 788

simplified setup 747, 757

ENCRYPTION, SET (policy rule) 736

enhancing  
network resiliency 189

ENOSPC error (strict replication 279)

Error messages  
certificates 835

establishing disaster recovery  
cluster 624

establishing quotas 431

EtherChannel 69

example

example (*continued*)  
    AFM relationship  
        GPFS protocol 929  
        NFS protocol 925  
example of encryption policy 740  
exceptions and limitations  
    applications considerations 473  
EXCLUDE 537  
EXCLUDE rule 531  
execute file permission 461  
exporting a GPFS file system 499  
extended attributes  
    Linux 474  
external lists  
    overview 577  
external pools  
    requirements 530  
external sink 129  
external storage pools  
    callbacks  
        lowDiskSpace 567  
        NO\_SPACE 567  
    defining 571  
    files  
        purging 575  
    managing  
        user-provided program 572  
migration 571, 574  
overview 530  
pre-migration 575  
recall 574  
requirements 530  
thresholds 567

## F

FEKs 735  
File  
    Compression 231  
file access 304  
file access frequency 579  
file access temperature 579  
file attributes  
    SQL expressions 541  
File attributes, testing for file encryption 561  
file audit logging  
    administering 911  
    configure 125  
    configuring 125  
    disabling 125  
    enabling 125, 126  
file system 125  
    GUI 128  
    manage 911  
    managing 911  
    message queue 126  
    mmaudit 125  
    mmmsgqueue 125  
file authentication  
    configure user authentication 323  
file clones  
    creating 613  
    deleting 615  
    listing 614

file clones (*continued*)  
    management 613  
    managing disk space 615  
    policy files 616  
    separating from parents 615  
    snapshots 615  
File encryption attribute, testing for 561  
file encryption keys 735  
File encryption, testing for file encryption attribute 561  
file heat 579  
file list file  
    format 572  
    record format 573  
file management  
    policies 531  
file permissions  
    control 462  
    GPFS extension 461  
file placement  
    policies 531  
File Placement Optimizer  
    configuring 688  
    distributing data 687  
    pool file placement and AFM 687  
    restrictions 734  
    upgrading 708  
file placement policy  
    default 531  
file reconciliations 882  
file replication  
    querying 230  
file system  
    acl 489  
    backup  
        SOBAR 617  
    ILM policy 598, 599  
    mount  
        GUI 219  
    mounting remote 512  
    permissions 693  
    pools  
        listing 528  
    remote access 512  
    restore  
        SOBAR 619  
    restoring  
        snapshot 604  
    set permissions 693  
    unmount  
        GUI 222  
file system acl 489  
file system configuration information, backing up  
    mmbbackupconfig command 250  
        using the mmbbackupconfig command 250  
file system maintenance mode 225  
file system manager  
    changing nodes 33  
    displaying node currently assigned 33  
    displaying nodes 33  
file system snapshots  
    subset restore 258  
file systems  
    access control lists 461  
    access from other cluster 505

file systems (*continued*)  
 access patterns of applications 69  
 AIX export 501  
 attributes  
     changing 229  
     displaying 228  
 backing up 244, 247  
 changing mount point on protocol nodes 220  
 checking 223  
 disk fragmentation 242  
 exporting 500, 981  
 exporting using NFS 499  
 file audit logging 128  
 format changes 267  
 format number 267  
 fragmentation  
     querying 242  
 granting access 516  
 Linux export 500  
 mounting on multiple nodes 218  
 physical connection 505  
 reducing fragmentation 243  
 remote access 516, 518  
 remote mount 512  
 remote mount concepts 505  
 repairing 223  
 restriping 240  
 revoking access 516  
 security keys 521, 522  
 snapshots 601  
 space, querying 241  
 unmounting on multiple nodes 221  
 user access 507  
 virtual connection 505

File-based configuration for performance monitoring tool 107

FileInherit 467

files  
 ./rhosts 66  
 /etc/group 507  
 /etc/passwd 507  
 /var/mmfs/ssl/id\_rsa.pub 513, 521  
 ill-placed 529  
 pre-migrating 575  
 storage pool assignment 528

files, stanza 202

fileset  
 /access/control 515  
 access control 515  
 creating fileset using GUI 589

Fileset creation 160

fileset quota 447

fileset snapshots  
 subset restore 259

filesets  
 attributes  
     changing 592  
 backing up 244, 245  
 block allocation 583  
 cautions 590  
 creating 588  
 deleting 590  
 dependent 582  
 in global snapshots 585

filesets (*continued*)  
 independent 582  
 inode allocation 583  
 linking 591, 592  
 managing 588  
 names 588  
 namespace attachment 582  
 overview 582  
 quotas 425, 583  
 root 582, 590, 592  
 snapshots 586  
 storage pool usage 584  
 unlinking 592  
 with mmbackup 586

FIPS compliance and encryption 860

FIPS1402mode 860

firewall considerations 995

firewall ports  
 examples of opening 996

firewall recommendations  
 call home 996  
 installation 985  
 NTP 985  
 protocols access 987  
 SKLM 992  
 Vormetric DSM 993

FlashCopy consistency groups 643

FOR FILESET 537

FPO  
 configuration changes 690  
 configuring 688  
 distributing data 687  
 pool file placement and AFM 687  
 restrictions 734  
 upgrading 708

FPO cluster 713

FPO clusters  
 administering 710  
 monitoring 710  
 monitoring, administering 710

FROM POOL 537

functions  
 CHAR 550  
 CONCAT 550  
 CURRENT\_DATE 551  
 CURRENT\_TIMESTAMP 551  
 DAY 551  
 DAYOFWEEK 551  
 DAYOFYEAR 551  
 DAYS 551  
 DAYSINMONTH 551  
 DAYSINYEAR 551  
 HEX 550  
 HOUR 552  
 INT 551  
 INTEGER 551  
 LENGTH 550  
 LOWER 550  
 MINUTE 552  
 MOD 551  
 MONTH 552  
 QUARTER 552  
 REGEX 550  
 REGEXREPLACE 550

functions (*continued*)

- SECOND 552
- SUBSTR 550
- SUBSTRING 551
- TIMESTAMP 552
- UPPER 551
- VARCHAR 551
- WEEK 552
- YEAR 552

## G

GDS 45

general considerations

- using storage replication 624

GPFS-based configuration

- integrate metrics with performance monitoring tool 106

GKLM 992

global snapshots

- with filesets 585

Google cloud storage 160

Google Cloud Storage

- Amazon S3 921

- GCS 921

GPFS

- adding CES groups in a cluster 61

- administering 666

- adminMode attribute 200

- CES packages

- deploying 58

- command principles 201, 202

- configuring 65, 66, 68–72

- configuring and tuning 90

- configuring CES 88

- configuring cluster 1, 58

- establishing disaster recovery 624

- File Placement Optimizer 683

- installing on Windows nodes 666

- managing cluster 1

- node quorum 30

- removing CES node 61

- removing protocol node 61

- shutting down cluster 36

- start 36

- stop 36

- tuning 65, 66, 68–72

GPFS administration security 66

GPFS cache 502

GPFS cluster

- adding nodes 4

- changing the GPFS cluster configuration servers 8

- creating 1

- deleting 5

- deleting nodes 5

- displaying configuration information 2

- managing 1

GPFS cluster configuration servers

- changing 8

- displaying 2

GPFS daemon

- starting 34

- stopping 35

GPFS file system

- administering 199

GPFS file system (*continued*)

- adminMode attribute 200

GPFS policy engine

- using 249

GPFS replication

- disaster recovery

- configuring 627, 629

- failback

- overview 631

- failback with permanent loss 633

- failback with temporary loss

- with CCR (Clustered Configuration Repository) 632

- failover

- overview 630

- gpfs\_iclose() 250

- gpfs\_iopen() 250

- gpfs\_iopen64() 250

- gpfs\_iread() 250

- gpfs\_ireaddir() 250

- gpfs\_ireaddir64() 250

- gpfs\_next\_inode() 250

- gpfs\_next\_inode64() 250

- gpfs\_open\_inodescan() 250

- gpfs\_open\_inodescan64() 250

- gpfs\_quotactl() 426

GPFS-specific

- mount options 218

- gpfs\_gskit 512

- GPU 45

- GPUDirect Storage 45

group id

- remapping 507

- group ID 507

- GROUP POOL 537

- group quota 447

GUI

- change GUI user password 454

- configure user authentication 323

- create users 453

- creating fileset 589

- creating nfs export 352

- creating SMB share 340

- creating snapshot 603

- define user permissions 453

- file audit logging 128

- firewall 991

- firewall recommendations 991

- limitations 999

- managing quotas 447

- mount file system 219

- port usage 984

- resume CES node 652

- start GPFS daemon 36

- sudo wrapper 30

- supported web browser settings 998

- supported web browser versions 998

- supported web browsers 998

- unmount file system 222

GUI administrators 451

GUI port usage, IBM Storage Scale 984

GUI user

- authentication 455

- multi-factor authentication 458

GUI users  
    change password [455](#)  
GUI web server  
    managing certificates [865](#)  
    security [865](#)

## H

Hadoop workloads  
    configuration [705](#)  
    tuning [705](#)  
handling  
    multiple nodes failure [720](#)  
handling node crashes [719](#)  
HAWC [971](#)  
HAWC, applications [971](#)  
HAWC, tuning and restrictions [972](#)  
HDFS protocol  
    Cluster Export Services (CES) [662](#)  
health  
    transparent cloud tiering service [906](#)  
Health status  
    Monitoring [179](#)  
helper threads  
    tuning [71](#)  
HEX [550](#)  
highly available write cache [971](#)  
highly-available write cache, applications [971](#)  
highly-available write cache, how to use. [972](#)  
highly-available write cache, tuning and restrictions [972](#)  
HighPercentage [539](#)  
hints, encryption [856](#)  
home cluster  
    NFS server [145](#)  
hostnames, changing [977](#)  
HOUR [552](#)  
HSM  
    firewall recommendations [995](#)

## I

IAM (integrated archive manager) modes  
    immutability [594](#)  
IBM ESS FlashCopy  
    disaster recovery [642](#)  
IBM Security Guardium Key Lifecycle Manager [992](#)  
IBM Spectrum Protect  
    backup scheduler [252](#)  
    configuration specifics  
        dsm.opt [255](#)  
        dsm.sys [253](#)  
    for IBM Spectrum Scale [253](#), [255](#)  
    scheduling backups [252](#)  
IBM Spectrum Protect backup planning  
    dsm.opt options [255](#)  
    dsm.sys options [253](#)  
IBM Spectrum Protect backup scheduler [252](#)  
IBM Spectrum Protect Backup-Archive  
    client  
    cautions  
        unlinking [592](#)  
IBM Spectrum Protect interface [247](#)  
IBM Spectrum Protect Manager backup planning [257](#)

IBM Spectrum Scale  
    Active File Management  
        tuning NFS client [169](#)  
        tuning NFS server [169](#)  
    Active File Management - Disaster Recovery [933](#)  
    Active File Management DR [149](#), [169](#), [933](#), [935](#), [936](#)  
    AFM  
        Administering [923](#)  
        creating AFM to cloud object storage relationship [940](#)  
        creating relationship [928](#)  
        gateway node [145](#)  
        NFS client [169](#)  
        NFS protocol [923](#)  
        NFS server [145](#)  
        setting up the cache cluster [925](#)  
        setting up the home cluster [923](#)  
        tuning NFS server on home/secondary cluster [170](#)  
        tuning\_gatewaynode [169](#)  
    AFM DR  
        Administering [933](#)  
        Changing gateway nodes in primary [149](#)  
        Changing NFS server at secondary [149](#)  
        Converting AFM relationship to AFM DR [936](#)  
        creating AFM-DR relationship [933](#)  
    AFM relationship  
        example [925](#)  
        GPFS protocol [929](#)  
    AFM to cloud object storage [940](#), [952](#), [954](#)  
    configuration [697](#)  
    Converting  
        GPFS filesets to AFM DR [935](#)  
    setting up  
        cache cluster [929](#)  
        home cluster [928](#)  
    tuning [697](#)  
    Tuning NFS backend  
        AFM [169](#)  
        AFM Dr [169](#)  
IBM Spectrum Scale for object storage  
    storage policies to fileset mapping [377](#)  
    storage policy for encryption [379](#)  
IBM Spectrum Scale GUI  
    snapshots [608](#)  
IBM Spectrum Scale Network Shared Disks  
    create [689](#)  
IBM Spectrum Scale NSD  
    create [689](#)  
IBM Spectrum Scale requirements [245](#)  
IBM Storage Protect  
    firewall considerations [995](#)  
IBM Storage Scale  
    access control lists  
        administration [465](#)  
        applying [471](#)  
        change [465](#), [472](#)  
        delete [465](#), [472](#)  
        display [471](#)  
        exceptions [472](#)  
        limitations [472](#)  
        setting [462](#), [470](#)  
        syntax [466](#)  
        translation [469](#)  
    access control lists (ACL)

IBM Storage Scale (*continued*)  
access control lists (ACL) (*continued*)  
    best practices 480  
    inheritance 479  
    permissions 482  
ACL administration 461  
activating quota limit checking 440  
active connections to SMB export 347  
Active File Management 133  
Add disks 276  
adding CES groups in a cluster 61  
adding node 4  
administering 666  
administering unified file and object access  
    example scenario 404  
AFM  
    configuration parameters 133  
apply ILM policy  
    transparent cloud tiering 877  
associate containers 401  
authentication  
    integrating with AD server 294  
authorizing protocol users 474, 475  
Backup 416  
CES IPs 53  
CES packages  
    deploying 58  
CES S3 359  
change GPFS disk states 281  
change GPFS parameters 281  
change NSD configuration 283  
change quota limit checking 443  
changing NFS export configuration 352  
changing Object configuration values 370  
changing the GPFS cluster configuration data 8  
check quota 436  
cloudkit administering 917–920  
cluster configuration information 2  
cluster quorum with tiebreaker 31  
configuring 65, 66, 68–72, 87  
configuring and tuning 90, 877  
configuring CES 88  
configuring cluster 1  
continuous replication of data 625  
create export on container 402  
create NFS export 351  
create SMB share ACLs 341  
Create SMB shares 343  
Creating SMB share 339  
creating storage policy 400  
data ingestion 410  
data integrity 624  
deactivating quota limit checking 441  
delete disk 276  
deleting node 5  
disaster recovery 634  
disaster recovery solutions 643  
disconnect active connections to SMB 348  
disk availability 280  
disk status 280  
Disks in a GPFS cluster 275  
display GPFS disk states 280  
enable file-access object capability 394  
Enable object access 403  
IBM Storage Scale (*continued*)  
establish and change quotas 431  
establishing  
    disaster recovery 624  
establishing disaster recovery 624  
export file systems 500  
file audit logging 125  
file system quota report  
    create 444  
file systems  
    AIX export 501  
firewall ports 994–996  
firewall recommendations 985, 987, 991, 996  
GPFS  
    mounting file system 917  
    unmounting file system 918  
GPFS access control lists (ACls)  
    manage 461  
GPFS cache usage 502  
GPFS quota management  
    disable 425  
    enable 425  
GPFS remote mount 917, 918  
How to manage OpenStack ACLs 372  
identity management modes for unified file and object access 386  
in-place analytics 407  
installing on Windows nodes 666  
limitations  
    transparent cloud tiering 909  
limitations of unified file and object access 408  
Linux export 500  
list NFS export 353  
list quota information 438  
list SMB shares 341  
local read-only cache 975  
Manage default quotas 427  
manage disk 275  
manage GPFS quotas 425  
manage GUI administrators 451  
Manage NFS exports 351, 353  
Manage S3 protocol 359  
Managing ACLs of SMB exports 345  
managing cloud storage tiers 877  
managing cluster 1  
managing protocol data exports 339  
managing SMB shares 339  
managing transparent cloud tiering service 906  
Mapping OpenStack commands  
    administrator commands 369  
migrating files  
    using transparent cloud tiering 880  
minimum release level 24  
Modifying SMB exports 344, 346  
MROT 189  
multi-region object deployment 381  
multi-region Swift Object deployment  
    adding region 380  
multiprotocol export considerations 356  
multiprotocol exports 356  
Network File System (NFS) 499  
NFS 500, 502  
NFS automount 503  
NFS export

IBM Storage Scale (*continued*)  
    NFS export (*continued*)  
        Unmount a file 502  
    NFS export configuration 503  
    node quorum 30  
    NSD server  
        Change server usage and failback 284  
    objectizer 391  
    Persistent Reserve (PR) functionality  
        disable 284  
        enable 284  
    point in time copy 643  
    protecting data 211  
    protocols disaster recovery 673  
    protocols DR 673  
    quotas  
        NFS 430  
        SMB 430  
    reconcile files  
        using transparent cloud tier 882  
    remote login 29  
    remote mounting 995  
    Remove NFS export 353  
    remove SMB shares 341  
    removing CES node 61  
    removing protocol node 61  
    replace disk 278  
    Restore 416  
    restore quota files 445  
    security mode 22  
    set quota 433  
    set up objectizer service interval 396  
    shutting down cluster 36  
    SMB and NFS protocols 356  
    SMB share configuration 340  
    storage policies for objects 377  
    storage-based replication 639  
    strict disk replication 279  
    sudo wrapper 26  
    sudo wrapper scripts 29  
    synchronous write operations 502  
    transparent cloud tiering service  
        managing 89, 906  
    tuning 65, 66, 68–72  
    unified file and object access 389, 393, 407  
    unified file and object access constraints 409  
    unified file and Swift Object access 382, 392  
    unified file and Swift Object access modes 384  
    use of consistency groups 624  
    using storage policy 400  
    vfs\_fruit support 288  
    view number of file locks in SMB export 350  
    view open files in SMP export 349  
    VLAN tagging 53

IBM Storage Scale cluster  
    create 688  
    creating 1  
    shutdown 36

IBM Storage Scale file attributes  
    modify 229

IBM Storage Scale file system  
    checking 223  
    create 690  
    repairing 223

IBM Storage Scale file system and pools  
    create 690

IBM Storage Scale file system attributes 228

IBM Storage Scale file systems  
    changing mount point on protocol nodes 220  
    deleting 222  
    management 217  
    mount options 218  
    mounting 217, 218  
    which nodes have mounted 223

IBM Storage Scale for object storage  
    administering storage policies 377  
    configuration files 412  
    EC2 credentials 371  
    managing object capabilities 373  
    S3 API 370  
    services 367  
    storage policy for compression 378  
    unified file and object access related user tasks 411

IBM Storage Scale for object versioning 374, 375

IBM Storage Scale FPO  
    configuration changes 690

IBM Storage Scale GUI port usage 984

IBM Storage Scale information units xxi

IBM Storage Scale license  
    apply 688

IBM Storage Scale log files 715

IBM Storage Scale make bulk changes NFS export 353

IBM Storage Scale port usage 982

IBM Storage Scale unmounting a file system 221

IBM TotalStorage  
    active-active cluster  
        configuration 635  
        failover 637  
    active-passive cluster  
        configuration 639  
        failover 641

ibmobjectizer service 391

ID mapping  
    shared authentication 390

identity management mode for unified file and object access  
    local\_mode 385

identity management modes unified file and object access  
    unified\_mode 386

identity management on Windows 665

ill-placed  
    files 529

ILM  
    creating policy 598, 599  
    ILM policy 598, 599  
    snapshot 578

ILM (information lifecycle management) 576

ILM (information lifecycle management)  
    overview 525

image backup 251

image restore 251

immutability  
    directories 594  
    effects 594  
    files 594  
    integrated archive manager (IAM) modes 594

immutable snapshots 211

import and export files 905

import files after upgrade to 5.2.1 [905](#)  
importing and exporting files [902](#)  
importing files exported by using old version of Cloud services [905](#)  
in-place analytics [407](#)  
independent fileset [515](#)  
InfiniBand  
    [IBV\\_EVENT\\_CQ\\_ERR](#) [913](#)  
    [IBV\\_WC\\_RNR\\_RETRY\\_EXC\\_ERR](#) [913](#)  
    Page pool [913](#)  
    verbsRdmaSend [913](#)  
    verbsRdmaPerConnection [913](#)  
    verbsRdmaPerNode [913](#)  
information lifecycle management (ILM)  
    overview [525](#)  
information lifecycle management(ILM) [576](#)  
inheritance flags [479](#)  
inheritance of ACLs  
    DirInherit [467](#)  
    FileInherit [467](#)  
    Inherited [467](#)  
    InheritOnly [467](#)  
Inherited [467](#)  
InheritOnly [467](#)  
installation  
    firewall [985](#)  
    firewall recommendations [985](#)  
installing GPFS, using mksysb [977](#)  
installing Windows IDMU [669](#)  
INT [551](#)  
INTEGER [551](#)  
integrate transparent cloud tiering metrics  
    with performance monitoring tool  
        using GPFS-based configuration [106](#)  
integrated archive manager (IAM) modes  
    immutability [594](#)  
integrating  
    transparent cloud tiering  
        performance monitoring tool [106](#)  
integrating transparent cloud tiering  
    performance monitoring tool [107](#)  
internal communication  
    port numbers [985](#)  
    recommended port numbers [985](#)  
internal communication among nodes  
    firewall [985](#)  
    firewall recommendations [985](#)  
    firewall recommendations for [985](#)  
internal storage pools  
    files  
        purging [575](#)  
    managing [526](#)  
    metadata [526](#)  
    overview [526](#)  
    system [526](#)  
    system.log [526](#)  
    user [526](#)  
IP addresses  
    CES (Cluster Export Services) [652](#)  
    private [518](#)  
    public [518](#)  
    remote access [518](#)  
IP addresses CNFS (Clustered Network environment) [646](#)  
IP addresses, changing [977](#)

ISV  
    multi-factor authentication [459](#)

## J

job  
    mmapplypolicy  
        phase 1 [553](#)  
        phase 2 [555](#)  
        phase 3 [557](#)  
Jumbo Frames [71](#)  
junction [582](#)

## K

Kafka [129](#)  
Kerberos based NFS access configuration  
    prerequisites [302](#)  
key cache purging, encryption [858](#)  
key clients  
    configurations [772](#)  
key manager for Cloud services [95](#)  
keys, encryption [735](#)

## L

large file systems  
    mmapplypolicy  
        performance [569](#)  
LDAP  
    bind user requirements [296](#)  
LDAP server [295](#)  
LDAP user information [298](#)  
LDAP-based authentication for file access  
    LDAP with Kerberos [318](#)  
    LDAP with TLS [316](#)  
    LDAP with TLS and Kerberos [319](#)  
    LDAP without TLS and Kerberos [320](#)  
LENGTH [550](#)  
level of functionality  
    minimum release level [24](#)  
LIMIT [538](#)  
limitations  
    cloud data sharing [909](#)  
    protocol support [511](#)  
Limitations  
    of the mmuserauth service create command [314](#)  
link aggregation [69](#)  
linking to  
    snapshots [606](#)  
Linux  
    CES (Clustered NFS) environment [645](#)  
listing  
    disks in storage pools [529](#)  
    file clones [614](#)  
    snapshots [603](#)  
listing exported files  
    using mmcloudmanifest tool [903](#)  
listing files  
    using transparent cloud tiering [885](#)  
lists  
    external [577](#)  
local read-only cache [975](#)

local read-only cache(LROC)  
  encryption [862](#)  
local snapshots  
  subset restore [258](#), [259](#)  
  subset restore using script [260](#)  
localityCopy [727](#)  
locked vault creation [112](#)  
locked vaults  
  creating [115](#)  
log files [715](#)  
lost+found directory [223](#)  
low-occupancy-percentage [539](#)  
LOWER [550](#)  
LTFS  
  firewall considerations [995](#)

## M

m4 macro processor  
  policy rules [562](#)  
Management GUI  
  supported web browsers [998](#)  
managing  
  a GPFS cluster [1](#)  
  filesets [588](#)  
  GPFS quotas [425](#)  
  GUI administrators [451](#)  
  multi-cluster protocols [510](#)  
  transparent cloud tiering service [89](#), [906](#)  
Managing  
  protocol services [287](#)  
managing cloud storage tiers  
  using IBM Storage Scale [877](#)  
managing disk space  
  file clones [615](#)  
managing multi-cluster protocol environment [510](#)  
manual  
  disk failure recovery [717](#)  
manual-based configuration for performance monitoring tool  
[107](#)  
manually  
  destroying files [884](#)  
MapReduce  
  create filesets for [691](#)  
  intermediate data [691](#)  
  intermediate data, temporary data [691](#)  
  temporary data [691](#)  
master encryption keys [735](#)  
master encryption keys (MEKs) [858](#)  
maxFilesToCache [981](#)  
maxFilesToCache parameter  
  definition [67](#)  
maxStatCache [981](#)  
maxStatCache parameter  
  definition [67](#)  
MEKs [735](#)  
memory  
  controlling [66](#)  
  swap space [70](#)  
  used to cache file data and metadata [67](#)  
memory considerations [71](#)  
metadata replication  
  changing [230](#)  
MIGRATE rule [531](#), [539](#)

migrating  
  warm data [880](#)  
migrating files  
  co-resident state [880](#)  
migrating files to the cloud storage tier [880](#)  
migration  
  external storage pools [571](#)  
MINUTE [552](#)  
miscellaneous SQL functions [552](#)  
mmaddcallback  
  diskIOErr [284](#)  
mmaddir [276](#), [527](#)  
mmaddnode [4](#), [979](#)  
mmapplypolicy  
  job  
    phase 1 [553](#)  
    phase 2 [555](#)  
    phase 3 [557](#)  
  overview [553](#)  
  performance  
    large file systems [569](#)  
mmaudit  
  file audit logging [125](#)  
mmaudit all list [911](#)  
mmauth [22](#), [505](#), [508](#), [512](#), [513](#), [516](#), [518](#), [520](#), [521](#)  
mmbackup  
  filesets [586](#)  
  firewall recommendations [995](#)  
MMBACKUP\_PROGRESS\_CALLOUT [248](#)  
mmbackupconfig [244](#)  
MMC  
  connect SMB exports [342](#)  
  connect SMB shares [342](#)  
  create SMB exports [343](#)  
  create SMB shares [343](#)  
  manage SMB export ACLs [345](#)  
  manage SMB exports [342](#)  
  manage SMB shares [342](#)  
  modify SMB exports [344](#)  
  remove SMB exports [344](#)  
  SMB export active connections [347](#)  
  SMB export disconnect connections [348](#)  
  SMB export offline settings [346](#)  
  SMB export open files [349](#)  
  SMB export view number of file locks [350](#)  
mmcallhome  
  detect system changes  
    use case [179](#)  
    use case [179](#)  
mmces [652](#), [654](#), [655](#)  
mmchattr [229](#)–[231](#), [240](#), [528](#)  
mmchcluster [8](#), [978](#)  
mmchconfig [22](#), [24](#), [501](#), [502](#), [505](#), [512](#), [518](#), [523](#), [651](#), [981](#)  
mmchconfig command [66](#)  
mmchdisk [240](#), [280](#), [281](#), [527](#)  
mmcheckquota [425](#), [436](#), [441](#), [443](#)  
mmchfileset [592](#)  
mmchfs [229](#), [284](#), [425](#), [440](#), [443](#)  
mmchnsd [978](#)  
mmchpolicy [531](#), [563](#)–[567](#), [736](#)  
mmcloudgateway destroy  
  command [884](#)  
mmcloudmanifest tool [903](#)  
mmcrcluster [1](#), [505](#)

mmcrfleset 588  
 mmcrfs 217, 425, 440, 465, 527  
 mmcrnsd 276  
 mmcsnapshot 250  
 mmdefragfs 242, 243  
 mmdelacl 465, 472  
 mmdeldisk 276, 527, 528  
 mmddelfleset 590  
 mmddelfs 222  
 mmdeinode 5, 979  
 mmdelesnapshot 607  
 mmdf 241, 276, 523, 526, 529  
 mmeditacl 465, 469, 470  
 mmexporta 425, 431  
 mmexportfs 981  
 mmfsck 276, 523  
 mmgetacl 462–464, 469–472  
 mmgetlocation 724  
 mmimports 981  
 mmlinkfleset 582, 588, 591, 592  
 mmllsattr 229, 230, 528, 582, 593  
 mmllscluster 2, 24, 514  
 mmllsconfig 523  
 mmllsdisk 280, 523, 981  
 mmllsfilesset 585, 586, 590, 593  
 mmllsfs 228, 279, 440, 441, 523, 528  
 mmllsmgr 33  
 mmllsmount 223, 523  
 mmllsnad 275, 978  
 mmllspolicy 566  
 mmllsquota 438  
 mmmount 217, 218, 284, 514  
 mmmsgqueue
 

- file audit logging 125

 mmnetverify
 

- command 215

 mmnfs export add command 351  
 mmnfs export change 353  
 mmnfs export load 353  
 mmobj command
 

- changing Object configuration values 370

 mmputacl 462, 464, 465, 470, 472  
 mmquotaff 440, 441  
 mmquotao 440  
 mmremotecluster 505, 514, 521, 523  
 mmremotes 284, 505, 514, 523  
 mmrepquota 444  
 mmrestorefs 586  
 mmrestripefile 528  
 mmrestripes
 

- completion time 204

 mmrpldisk 527  
 mmsetquota 425  
 mmshutdown 35, 651, 977, 978  
 mmsmb
 

- list SMB shares 341

 mmsnapdir 586  
 mmstartup 35, 651, 978  
 mmumount 221  
 mmunlinkfleset 582, 590, 592  
 mmuserauth 293, 300, 303, 314  
 mmwatch
 

- command 129
- enable 129

mmwatch (*continued*)
 

- steps 129
- watch 129

 mmwatch command 129  
 MOD 551  
 modifying file system attributes 229  
 MONTH 552  
 mount file system
 

- GUI 219

 mount problem
 

- remote cluster 523

 mounting
 

- file systems 217
- GPFS remote mount 917

 mounting a file system
 

- an NFS exported file system 499

 multi-cluster environments
 

- upgrade 510

 multi-cluster protocol environment
 

- IBM Storage Scale 510

 Multi-Rail over TCP
 

- network resiliency 189

 multi-region object deployment
 

- administering 381
- exporting configuration data 381
- importing configuration data 381
- removing region 381

 multi-region Swift Object deployment
 

- adding region 380

 multicuster
 

- file system access 505

 Multiple nodes failure without SGPanic 720  
 multiple subnets 189  
 multiple versions of data
 

- IBM Storage Scale 624

 Multiprotocol export considerations
 

- NFS export 356
- SMB export 356

## N

Network configuration  
 CES (Cluster Export Services) 652  
 Network File System (NFS)
 

- cache usage 502
- exporting a GPFS file system 499
- interoperability with GPFS 499
- synchronous writes 502
- unmounting a file system 502

 Network Information Server 322  
 network interfaces 69  
 Network Shared Disks
 

- create 689

 Network Shared Disks (NSDs)
 

- changing configuration attributes 283

 network switch failure 721  
 nfs
 

- creating 352

 NFS
 

- quotas 430

 NFS automount 503  
 NFS export
 

- create NFS export 351
- list NFS export 353

NFS export (*continued*)  
     make NFS export change 353

NFS exports  
     Manage NFS exports  
         GUI navigation 353

NFS protocol  
     Cluster Export Services (CES) 656  
     creating an AFM relationship 923

NFS protocol disaster recovery  
     failback steps 678  
     failover steps 678

NFS protocol DR  
     failback steps 678  
     failover steps 678

NFS protocol services  
     starting 287

NFS V4 461

NFS V4 ACL  
     exceptions and limitations 473  
     special names 473

NFS V4 protocol  
     exceptions and limitations 473

NFS/SMB protocol over remote cluster mounts 508

NFSv4 based access  
     authentication considerations 301

NIS-based authentication for file access 322

NIST compliance 522

NIST compliance and encryption 861

node classes, user-defined 201

node crash 719

node failure 718

node numbers, changing 977

node quorum 30

node quorum with tiebreaker 7

node state 715

nodeJoin Event 733

nodeLeave Event 733

nodes  
     adding to a GPFS cluster 4  
     assigned as file system manager 33  
     firewall 985  
     renaming or renumbering 977  
     specifying with commands 201  
     swap space 70  
     which have file systems mounted 223

NSD  
     create 689

NSD fallback 284

NSD local disks  
     diskIOErr 284  
     periodic check 284

NSD server 204, 284, 505

NSD server list  
     changing 283

NSD server nodes  
     changing 283

NSD stanza 202

NVIDIA 45

**O**

OBJ protocol  
     Cluster Export Services (CES) 659

object

object (*continued*)  
     network groups 420  
     node 420

object access  
     enabling 382  
     existing filesets 382

object capabilities  
     disabling 373  
     enabling 373  
     listing 373  
     managing 373

object Configuration values  
     Changing 370

object heatmap policy  
     enabling 422

object storage  
     backup 416, 420  
     containers 491  
     creating containers 491  
     restore 418

Object storage  
     Backup 416  
     Restore 416

object storage services  
     managing 367

object versioning  
     disabling 375  
     enabling 374  
     example 375  
     managing 374

objectization 391

objectizer 391

Open Stack components 183

OpenLDAP  
     server ACLs 296

OpenStack ACLs  
     how to manage 372  
     using S3 API 372

OpenStack commands  
     Mapping 369

OpenStack EC2 credentials  
     configuring 371

OpenWrite and OpenRead rule  
     transparent cloud tiering 877

Operating system  
     configuration 694  
     tuning 694

Optional  
     configuration 703  
     tuning 703

options  
     always 219  
     asfound 219  
     asneeded 219  
     atime 218  
     mtime 218  
     never 219  
     noatime 218  
     nomtime 218  
     norelatime 219  
     nosyncnfs 219  
     relatime 219  
     syncnfs 219  
     useNSDserver 219

Oracle  
   GPFS use with, tuning [72](#)  
 orphaned files [223](#)  
 overlapping unixmap ranges [312](#)  
 owning cluster [128](#)

**P**

packages  
   gpfs.gskit [512](#)

Page pool  
   InfiniBand [913](#)  
   Large [913](#)

pagepool parameter  
   usage [66](#)

parents  
   file clones [615](#)

password  
   change GUI user password [454](#)

password policy [454](#)

performance  
   access patterns [69](#)  
   aggregate network interfaces [69](#)  
   disk I/O settings [72](#)  
   mmapplypolicy [568](#)  
   monitoring using mmpmon [65](#)  
   setting maximum amount of GPFS I/O [72](#)

Performance Monitoring tool  
   firewall [994](#)

performing  
   rolling upgrade [711](#)

Persistent Reserve  
   disabling [284](#)  
   enabling [284](#)

physical disk stanza [202](#)

physically broken disks [718](#)

policies  
   assigning files [563](#)  
   changing active [565](#)  
   creating [563](#)  
   default [567](#)  
   default storage pool [563](#)  
   deleting [567](#)  
   error checking [531](#)  
   external storage pools  
     managing [563](#)  
   file management [531](#)  
   file placement [531, 585](#)  
   installing [564](#)  
   listing [566](#)  
   overview [531](#)  
   policy rules [533](#)  
   SET POOL [563](#)  
   validating [566](#)

policies, encryption [736](#)

policies, rewrapping [741](#)

policy  
   creating [598, 599](#)

policy example, encryption [740](#)

policy files  
   file clones [616](#)

policy rule, ENCRYPTION IS [736](#)

policy rule, SET ENCRYPTION [736](#)

policy rules  
   built-in functions  
     date and time [546](#)  
     extended attribute [546](#)  
     miscellaneous [546](#)  
     numerical [546](#)  
     string [546](#)  
   DELETE [575](#)  
   examples [557](#)  
   EXTERNAL POOL [574](#)  
   m4 macro processor [562](#)  
   overview [533](#)  
   SQL expressions in [540](#)  
   syntax [534](#)  
   terms [535](#)  
   tips [557](#)  
   types [534](#)

policy rules, encryption [736](#)

pools, external  
   requirements [530](#)

port usage, IBM Storage Scale [982](#)

PR [284](#)

pre-migrating files  
   cloud storage tier [880](#)

pre-migration  
   overview [575](#)

prefetchThreads parameter  
   tuning  
     on Linux nodes [71](#)  
     use with Oracle [72](#)

preparations for SOBAR [890](#)

prerequisite  
   Kerberos-based SMB access [301](#)

prerequisites  
   Kerberos based NFS access [302](#)  
   LDAP server [295](#)

prerequisites and preparations for SOBAR (Cloud services)  
   primary site [890](#)  
   recovery site [891](#)

primary site preparations for SOBAR [890](#)

principles  
   common to GPFS commands [201, 202](#)

Procedure for SOBAR (Cloud services) [889](#)

protecting  
   data [211](#)

protection of data [735](#)

protocol data  
   security [867](#)

protocol data security  
   protocol, data security [869](#)

protocol node  
   remove from cluster [61](#)

protocol nodes  
   firewall [984](#)  
   firewall recommendations [984](#)

protocol over remote cluster mounts [508](#)

protocol support on remotely mounted file system  
   limitations [511](#)

protocols  
   administration tasks [49, 60](#)  
   removal tasks [49](#)

protocols access  
   CES IP aliasing [53](#)  
   port usage [987](#)

protocols data exports 339  
protocols disaster recovery  
  authentication configuration 679  
  authentication configuration fallback 680  
  authentication configuration failover 679  
  authentication configuration restore 680  
CES 680  
CES configuration 680  
configuration information  
  collecting 675  
example setup 674  
gateway node 674  
gateway node setup 674  
limitations 673  
NFS protocol data 677  
overview 673  
prerequisites 673  
SMB protocol data 676

protocols DR  
  authentication configuration 679  
  authentication configuration fallback 680  
  authentication configuration failover 679  
  authentication configuration restore 680  
CES 680  
CES configuration 680  
configuration information  
  collecting 675  
example setup 674  
gateway node 674  
gateway node setup 674  
limitations 673  
NFS protocol data 677  
overview 673  
prerequisites 673  
SMB protocol data 676

protocols nodes  
  CES IP aliasing 53

protocols on a remotely mounted cluster  
  configuring 509

protocols over remote cluster mounts 508

purging, encryption key cache 858

## Q

QoS  
  support for autorecovery 734

QoS Classes  
  maintenance 237  
  other 237

Quality of Service for I/O operations  
  (QoS)  
  configuring 237

QUARTER 552

querying  
  disk fragmentation 242  
  file system fragmentation 242  
  replication 229  
  space 241

Querying file system 241

quota files  
  backing up 445  
  restoring 445

quotas  
  activating limit checking 440

quotas (*continued*)  
  changing 431  
  changing limit checking 443  
  checking 436  
  creating reports 444  
  deactivating limit checking 441  
  default values 427  
  disabling 425  
  displaying 438  
  enabling 425  
  establishing 431  
  fileset 425  
  group 425  
  managing quotas using GUI 447  
  user 425

## R

RDMA tuning 913  
read file permission 461  
read-only cache, local 975  
rebalancing  
  storage pools 529  
reboot node intentionally 718  
recall files  
  from cloud storage tier 882  
reconciliations of files  
  IBM Storage Scale 882  
record format  
  file list file 573  
recover a node manually 718  
recover node automatically 719  
recover node manually 719  
recovery  
  cluster 623  
recovery group stanza 202  
recovery node automatically 718  
recovery site preparations for SOBAR 891  
Red Hat Open Stack Platform 184  
Redundant Array of Independent Disks (RAID)  
  RAID5 performance 72

REGEX 550

REGEXREPLACE 550

remapping  
  group id 507  
  user id 507

remote 128

remote access

  AUTHONLY 520  
  displaying information 523  
  encrypted file 774, 778, 807  
  file system 512  
  IP addresses 518  
  managing 516  
  mount problem 523  
  restrictions 523  
  security keys 521  
  security levels 520  
  updating 523

remote cluster

  displaying information 523  
  mount problem 523  
  restrictions 523  
  updating 523

remote key management server setup [742](#)  
remote mounting  
  firewall considerations [995](#)  
remotely  
  mounted [128](#)  
Renew certificates [835](#), [845](#)  
repairing  
  file system [223](#)  
replace broken disks [731](#)  
replace more than one active disks [731](#)  
replacing disks [278](#), [279](#)  
REPLICATE [538](#)  
REPLICATE clause [538](#)  
replication  
  changing [229](#), [230](#)  
  querying [229](#)  
  storage pools [529](#)  
  system storage pool [526](#)  
replication of data  
  IBM Storage Scale [625](#)  
requirements  
  administering GPFS [199](#)  
  external pools [530](#)  
  for IBM Spectrum Scale [245](#)  
requirements (setup), encryption [742](#)  
restarting  
  IBM Storage Scale cluster [713](#)  
restore  
  file system  
    SOBAR [619](#)  
  storage pools [576](#)  
restore option  
  transparent cloud tiering [886](#)  
Restore option  
  Cloud services configuration [887](#)  
restore procedure (using SOBAR) [894](#)  
restore/backup and encryption [862](#)  
Restoring  
  Cloud services configuration [887](#)  
Restoring deleted files  
  from cloud storage tier [886](#)  
Restoring files  
  transparent cloud tiering [886](#)  
restoring from local snapshots  
  using the sample script [260](#)  
restoring the locality for files  
  with WADFG [729](#)  
  without WADFG [728](#)  
restrictions and tuning for highly-available write cache [972](#)  
restriping a file system [240](#)  
resume CES node [652](#)  
revoking  
  old certificate [117](#)  
rewrapping policies [741](#)  
RKM back ends [742](#)  
RKM server setup [742](#)  
rolling upgrades [711](#)  
root authority [66](#)  
root fileset [590](#), [592](#)  
root squash [508](#)  
root squashing [507](#)  
root-level processes  
  sudo wrappers [30](#)  
rotating

rotating (*continued*)  
  client key [117](#)  
rule (policy), ENCRYPTION IS [736](#)  
rule (policy), SET ENCRYPTION [736](#)  
RULE clause  
  ACTION [535](#)  
  COMPRESS [535](#)  
  DIRECTORIES\_PLUS [536](#)  
  EXCLUDE [537](#)  
  FOR FILESET [537](#)  
  FROM POOL [537](#)  
  GROUP POOL [537](#)  
  LIMIT [538](#)  
  REPLICATE [538](#)  
  SET POOL [538](#)  
  SHOW [539](#)  
  THRESHOLD [539](#)  
  TO POOL [540](#)  
  WEIGHT [540](#)  
  WHEN [540](#)  
  WHERE [540](#)  
rules, encryption policy [736](#)

## S

S3 ACLs  
  how to manage [372](#)  
S3 API  
  enabling [370](#)  
S3 protocol services  
  starting [287](#)  
S3 services  
  Manage S3 accounts and buckets [359](#)  
S3 user access [359](#)  
safeguarded copy  
  snapshots [211](#)  
samba attributes [298](#)  
Scale out back and restore procedure (Cloud services) [889](#)  
Scale Out Backup and Restore [251](#)  
scale out backup and restore (SOBAR)backup [617](#)  
scale out backup and restore (SOBAR)overview [617](#)  
scale out backup and restore (SOBAR)restore [619](#)  
scaling out  
  cloud cluster [918](#)  
scheduling the maintenance activities [100](#)  
script  
  external pool [574](#)  
SECOND [552](#)  
secure deletion of data [735](#)  
secure deletion, encryption [856](#)  
secure protocol data [867](#)  
security  
  firewall recommendations [984](#)  
security key  
  changing [522](#)  
security keys  
  remote access [521](#)  
security levels  
  AUTHONLY [520](#)  
  cipherList [520](#)  
  remote access [520](#)  
security mode  
  managing remote access [22](#)  
Security Token Service [159](#)

security, administration 66  
selective objectization 403  
Server  
    certificates 838  
server setup, RKM 742  
SET ENCRYPTION policy rule 736  
set file system permissions 693  
SET POOL 538  
set up a private certificate 113  
set up a private key 113  
set up authentication servers  
    integrating with AD server 294  
    integrating with LDAP server 295  
Setting  
    LDAP server prerequisites 295  
setting access control lists 462  
setting and changing the immutability of files  
    effects of file operations on immutable files 595  
    immutability restrictions 594  
setting private key and locked vaults 112  
setting quotas  
    per-project 433  
setting up  
    cache cluster 929  
    home cluster 928  
setting up a maintenance window 100  
setting up a private key 112  
setting up Transparent cloud tiering for WORM 112  
Setting up Transparent cloud tiering on a remotely mounted  
file system 109  
setup requirements, encryption 742  
setup, RKM server 742  
SGPanic for handling node failure 721  
shared root directory)  
    CES (Cluster Export Services) 651  
SHOW 539  
shutdown cluster 36  
shutting down  
    cluster 36  
simplified tasks  
    encryption 780  
sink 129  
skipping  
    filesets 126  
SKLM  
    firewall recommendations 992  
SKLM: Certificate chain 761, 797, 810  
SKLM: Regular setup with certificate chain 797, 810  
SKLM: Simplified setup with certificate chain 761  
SMB  
    creating a share 340  
    quotas 430  
SMB exports  
    active connections 347  
    connecting 342  
    creating 343  
    disconnect connections 348  
    managing 342  
    managing ACLs 345  
    modifying 344  
    offline settings 346  
    removing 344  
    view number of file locks 350  
    view open files 349  
SMB protocol  
    Cluster Export Services (CES) 658  
SMB protocol disaster recovery  
    failback steps 677  
    failover steps 676  
SMB protocol DR  
    failback steps 677  
    failover steps 676  
SMB protocol services  
    starting 287  
SMB shares  
    active connections 347  
    connecting 342  
    creating 343  
    disconnect connections 348  
    GUI navigation 339  
    managing 342  
    managing ACLs 345  
    managing SMB shares 339  
    modifying 344  
    offline settings 346  
    removing 344  
    view number of file locks 350  
    view open files 349  
SNAP\_ID 552  
snapshot  
    creating 603  
snapshots  
    creating 601  
    deleting 607  
    file clones 615  
    file system restoring 604  
    IBM Spectrum Scale GUI 608  
    linking to 606  
    listing 603  
    overview 601  
    reading mmapplypolicy 605  
    safeguarded copy 211  
snapshots and encryption 862  
snapshots, fileset 586  
snapshots, global  
    with filesets 585  
SOBAR 251  
SOBAR (Scale out backup and restore)backup 617  
SOBAR (Scale out backup and restore)overview 617  
SOBAR (Scale out backup and restore)restore 619  
SOBAR backup prerequisites  
    primary site 890  
    recovery site 891  
SOBAR restore procedure  
    Cloud services 894  
SparkWorkloads  
    configuration 706  
    tuning 706  
spectrumscale 60  
SQL  
    expressions  
        file attributes 541  
        in policy rules 540  
SQL expressions  
    file attributes 541  
    in policy rules 540  
SQL functions  
    SNAP\_ID 552

SQL functions, miscellaneous  
     functions, miscellaneous SQL [552](#)  
 standards compliance  
     encryption [860](#)  
 stanza files [202](#), [527](#)  
 stanza, declustered array [202](#)  
 stanza, NSD [202](#)  
 stanza, physical disk [202](#)  
 stanza, recovery group [202](#)  
 stanza, virtual disk [202](#)  
 start CES node [652](#)  
 start GPFS daemon [36](#)  
 starting  
     Cloud services [89](#)  
     transparent cloud tiering service [89](#)  
 starting and stopping ibmobjectizer [395](#)  
 starting GPFS  
     before starting [34](#)  
 status  
     disk [280](#)  
 steps for SOBAR in Cloud services [889](#)  
 stop CES node [652](#)  
 stop GPFS daemon [36](#)  
 stopping  
     transparent cloud tiering service [906](#)  
 stopping GPFS [34](#), [35](#)  
 storage  
     partitioning [525](#)  
 storage management  
     automating [525](#)  
     tiered [525](#)  
 storage policies [377](#)–[379](#)  
 storage policies for object  
     administering [377](#)  
     compression [378](#)  
     encryption [379](#)  
     mapping to filesets [377](#)  
 storage pools  
     backup [576](#)  
     creating [527](#)  
     deleting [528](#)  
     disk assignment  
         changing [527](#)  
     external  
         working with [570](#)  
     file assignment [528](#)  
     files  
         listing fileset of [528](#)  
         listing pool of [528](#)  
     listing [528](#)  
     listing disks in [529](#)  
     managing [527](#)  
     names [527](#)  
     overview [525](#)  
     rebalancing [529](#)  
     replication [529](#)  
     restore [576](#)  
     subroutines  
         `gpfs_fgetattr()` [576](#)  
         `gpfs_fputattr()` [576](#)  
         `gpfs_fputattrswithpathname()` [576](#)  
     system storage pool [526](#)  
     system.log storage pool [527](#)  
     user storage pools [579](#)  
 storage replication  
     general considerations [624](#)  
 storage-base replication  
     synchronous mirroring [634](#)  
 Strict replication  
     Disk offline [279](#)  
 subnet [518](#)  
 subnets  
     MROT [189](#)  
 subroutines  
     `gpfs_iclose()` [250](#)  
     `gpfs_iopen()` [250](#)  
     `gpfs_iopen64()` [250](#)  
     `gpfs_iread()` [250](#)  
     `gpfs_ireaddir()` [250](#)  
     `gpfs_ireaddir64()` [250](#)  
     `gpfs_next_inode()` [250](#)  
     `gpfs_next_inode64()` [250](#)  
     `gpfs_open_inodescan()` [250](#)  
     `gpfs_open_inodescan64()` [250](#)  
     `gpfs_quotactl()` [426](#)  
 SUBSTR [550](#)  
 SUBSTRING [551](#)  
 sudo wrapper [26](#)  
 sudo wrapper scripts  
     configuring on existing cluster [29](#)  
     configuring on new cluster [29](#)  
 sudo wrappers  
     root-level processes [30](#)  
 suspend CES node [652](#)  
 swap space [70](#)  
 Swift Object protocol service  
     starting [290](#)  
 synchronous mirroring  
     GPFS replication [625](#)  
     using storage-base replication [634](#)  
 syntax  
     policy rules [534](#)  
 system storage pool  
     deleting [528](#)  
     highly reliable disks [526](#)  
     replication [526](#)  
 system.log pool  
     deleting [528](#)  
 system.log storage pool  
     definition [527](#)

## T

TCP window [71](#)  
 tenants  
     configurations [772](#)  
 terms  
     policy rules [535](#)  
 THRESHOLD [539](#)  
 tiering data based on file access frequency [579](#)  
 TIMESTAMP [552](#)  
 Tivoli directory server  
     ACLs [297](#)  
 TO POOL [540](#)  
 transparent cloud tiering  
     administering files [877](#)  
     automatically applying a policy [877](#)  
     clean up files [883](#)

transparent cloud tiering (*continued*)  
database recovery 888  
dmremove commands 883  
limitations 909  
listing files migrated to the cloud 885  
migrating files 880  
recall files 882  
ZIMon integration 106

Transparent cloud tiering  
cloud storage access points 92  
configuration 92  
creating a key manager 95  
creating container pairs 96  
listing exported files 903

Transparent Cloud Tiering  
cloud data sharing 902  
importing and exporting files 902

Transparent cloud tiering configuration 94

transparent cloud tiering service  
health monitoring 906  
managing 906  
starting 89  
stopping 906

tuning  
gateway node 169  
NFS client 169  
NFS server 169, 170  
NFS server on the home/secondary cluster 170  
transparent cloud tiering 87

tuning and restrictions for highly-available write cache 972

Tuning NFS backend  
AFM 169  
AFM DR 169

tuning parameters  
prefetch threads  
on Linux nodes 71  
use with Oracle 72  
worker threads  
on Linux nodes 71  
use with Oracle 72

## U

UID remapping 508

unified file and object access  
administering 394  
associating container 401  
authentication 389  
configuration files 412  
configuring authentication 398  
constraints 409  
creating NFS export 402  
creating SMB export 402  
creating storage policy 400  
data ingestion through object 410  
example scenario 404  
examples 410  
file path  
determining 393  
file-access capability 394  
identity management modes 398  
limitations 408  
object path  
determining 393

unified file and object access (*continued*)  
object-server-sof.conf 412  
objectization 391  
objectizer service interval 396  
POSIX path  
determining 393  
scheduling objectizer 396  
selective objectization 403  
setting up mode 398  
spectrum-scale-object.conf 412  
spectrum-scale-objectizer.conf 412  
unified\_mode identity management 386  
use cases 407

unified file and object access related user tasks  
curl commands 411

unified file and object access storage policy  
associate container 401  
creating export 402

unified file and Swift Object access  
administering 382  
identity management modes 384  
managing 382  
POSIX path 392  
Swift Object path 392

unified file and Swift Object access modes 384

unified file and Swift ObjectSwift Object access  
file path 392

umount file system  
GUI 222

unmounting  
GPFS remote mount 918

unmounting a file system  
NFS exported 502  
on multiple nodes 221

update  
new key and certificate 119

updatedb considerations 71

updating a cloud storage account 90

updating a CSAP 92

updating new key and certificate 119

upgrade multi-cluster environments  
IBM Storage Scale 510

upgrading  
applying maintenance to your gpfs system 225

upgrading other infrastructure 713

UPPER 551

use case for configuring call home 176

use of consistency groups  
point in time copy 643

useNSDserver  
values 219

user account 507

user id  
remapping 507

user ID 507

user quota 447

user storage pool  
deleting 528

user storage pools  
access temperature 579  
data blocks 579

user-defined node classes 201

user-provided program  
external storage pools 572

users  
    change password 455  
using a clustered NFS subsystem 503  
using highly-available write cache 972  
using protocol over remotely mounted file system 508  
using the GPFS policy engine 249

## V

validation of descriptors on disk dynamically 225  
VARCHAR 551  
vfs\_fruit support  
    SMB protocol 288  
virtual disk stanza 202  
Vormetric DSM  
    firewall recommendations 993

## W

warning  
    certificate expiration 832  
WEEK 552  
WEIGHT 540  
WHEN 540  
WHERE 540  
Windows  
    auto-generated ID mappings 665  
    Configuring ID mappings in Active Directory Users and Computers 666  
    configuring ID mappings in IDMU 670  
    identity management 665  
    IDMU installation 669  
    installing IDMU 669  
workerThreads parameter  
    tuning  
        on Linux nodes 71  
        use with Oracle 72  
WORM solution  
    deploying 111  
WORM solutions  
    deploying 115  
    deployment 110  
        set up Transparent cloud tiering 112  
write cache, highly available 971  
write file permission 461  
write once read many  
    solutions 110

## Y

YEAR 552





Product Number: 5641-DM1  
5641-DM3  
5641-DM5  
5641-DA1  
5641-DA3  
5641-DA5  
5737-F34  
5737-I39  
5765-DME  
5765-DAE

SC28-3907-01

