

IBM Spectrum Scale
5.1.5

*Concepts, Planning, and Installation
Guide*



Note

Before using this information and the product it supports, read the information in [“Notices” on page 595](#).

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

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

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 xxxii. 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, 2022.**

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

Contents

Figures.....	xi
Tables.....	xiii
About this information.....	xv
Prerequisite and related information.....	xxxi
Conventions used in this information.....	xxxi
How to send your comments.....	xxxii
Summary of changes.....	xxxv
Chapter 1. Introducing IBM Spectrum Scale.....	1
Overview of IBM Spectrum Scale.....	1
Strengths of IBM Spectrum Scale.....	1
Basic structure of IBM Spectrum Scale.....	4
IBM Spectrum Scale cluster configurations.....	6
GPFS architecture.....	8
Special management functions.....	8
Use of disk storage and file structure within a GPFS file system.....	11
GPFS and memory.....	14
GPFS and network communication.....	15
Application and user interaction with GPFS.....	18
NSD disk discovery.....	23
Failure recovery processing.....	24
Cluster configuration data files.....	25
GPFS backup data.....	26
Cluster configuration repository.....	26
GPUDirect Storage support for IBM Spectrum Scale.....	26
Protocols support overview: Integration of protocol access methods with GPFS.....	28
Cluster Export Services overview.....	29
NFS support overview.....	31
SMB support overview.....	32
Object storage support overview.....	33
Active File Management.....	38
Introduction to Active File Management (AFM).....	39
Overview and concepts.....	39
Active File Management (AFM) features.....	53
AFM limitations.....	94
AFM-based Asynchronous Disaster Recovery (AFM DR)	96
Introduction.....	97
Recovery time objective (RTO).....	98
Modes and concepts.....	99
AFM-based Asynchronous Disaster Recovery features.....	99
AFM DR limitations.....	106
AFM DR deployment considerations and best practices.....	109
AFM to cloud object storage.....	117
AFM to cloud object storage operation modes.....	119
AFM to cloud object storage parallel read data transfer.....	125
Connectivity to cloud object storage.....	128

Eviction in AFM to cloud object storage.....	129
AFM to cloud object storage limitations.....	129
Audit messages support for the AFM to cloud object storage.....	131
Partial file or object caching for AFM to cloud object storage.....	131
AFM to cloud object storage directory object support.....	134
AFM to cloud object storage support for more than 2 K metadata.....	135
AFM to cloud object storage policy based upload for manual updates mode	137
Support of Google cloud storage platform for AFM to cloud object storage.....	139
Symbolic links.....	140
Introduction to system health and troubleshooting	140
Introduction to performance monitoring.....	142
Data protection and disaster recovery in IBM Spectrum Scale.....	143
Data backup options in IBM Spectrum Scale.....	143
Data restore options in IBM Spectrum Scale.....	144
Data mirroring in IBM Spectrum Scale.....	144
Protecting file data using snapshots	144
Introduction to Scale Out Backup and Restore (SOBAR).....	144
Commands for data protection and recovery in IBM Spectrum Scale.....	144
IBM Spectrum Scale GUI.....	145
IBM Spectrum Scale management API.....	149
Functional overview.....	150
API requests.....	151
API responses.....	155
Asynchronous jobs.....	157
Accessing the IBM Spectrum Scale REST API endpoint details through Swagger and API explorer.....	158
List of IBM Spectrum Scale management API commands.....	161
Cloud services	168
How Transparent cloud tiering works.....	170
How Cloud data sharing works.....	171
How Write Once Read Many (WORM) storage works.....	174
Supported cloud providers.....	175
Interoperability of Transparent cloud tiering with other IBM Spectrum Scale features.....	175
Interoperability of Cloud data sharing with other IBM Spectrum Scale features.....	177
File audit logging.....	178
Producers in file audit logging.....	179
The file audit logging fileset.....	179
File audit logging records.....	179
File audit logging events.....	180
JSON attributes in file audit logging.....	181
Remotely mounted file systems in file audit logging.....	183
Clustered watch folder.....	184
Producers in clustered watch folder.....	184
Interaction between clustered watch folder and the external Kafka sink.....	184
Clustered watch folder events.....	185
JSON attributes in clustered watch folder.....	186
Understanding call home	188
Types of call home data upload.....	191
Inspecting call home data uploads.....	206
Benefits of enabling call home.....	207
Data privacy with call home.....	208
Call home monitors for PTF updates.....	209
IBM Spectrum Scale in an OpenStack cloud deployment.....	210
IBM Spectrum Scale product editions.....	213
IBM Spectrum Scale license designation.....	215
Capacity-based licensing.....	218

Chapter 2. Planning for IBM Spectrum Scale.....	219
Planning for GPFS.....	219
Hardware requirements	219
Software requirements.....	220
Recoverability considerations.....	223
GPFS cluster creation considerations.....	228
Disk considerations.....	238
File system creation considerations.....	251
Backup considerations for using IBM Spectrum Protect.....	270
Planning for Quality of Service for I/O operations (QoS).....	282
Planning for extended attributes.....	283
Planning for the Highly Available Write Cache feature (HAWC).....	286
Planning for systemd.....	289
Planning for protocols.....	290
Authentication considerations.....	291
Planning for NFS.....	301
Planning for SMB.....	305
SMB best practices.....	310
Fileset considerations for creating protocol data exports.....	315
Planning for Object Storage deployment.....	317
Planning for CES HDFS.....	329
Planning for Cloud services	329
Hardware requirements for Cloud services	329
Software requirements for Cloud services	329
Network considerations for Cloud services	331
Cluster node considerations for Cloud services	331
IBM Cloud Object Storage considerations.....	332
Firewall recommendations for Cloud services.....	334
Performance considerations.....	335
Security considerations.....	338
Planning for maintenance activities.....	339
Backup considerations for Transparent cloud tiering.....	340
Quota support for tiering.....	341
Client-assisted recalls.....	341
Planning for AFM.....	341
Requirements for UID and GID on the cache and home clusters.....	341
Recommended workerThreads on a cache cluster.....	342
Inode limits to set at cache and home.....	342
Planning for AFM gateway nodes.....	342
General recommendations for AFM gateway node configuration.....	343
Planning for AFM DR.....	343
Requirements for UID/GID on primary and secondary clusters.....	343
Recommended worker1Threads on primary cluster.....	343
NFS setup on the secondary cluster.....	344
General guidelines and recommendations for AFM-DR.....	344
Planning for AFM to cloud object storage.....	345
Firewall recommendations for AFM to cloud object storage.....	346
Firewall recommendations.....	346
Considerations for GPFS applications.....	346
Security-Enhanced Linux support.....	346
Space requirements for call home data upload.....	347
Chapter 3. Steps for establishing and starting your IBM Spectrum Scale cluster..	349
Chapter 4. Installing IBM Spectrum Scale on Linux nodes and deploying protocols.....	351

Deciding whether to install IBM Spectrum Scale and deploy protocols manually or with the installation toolkit.....	351
Installation prerequisites.....	352
Preparing the environment on Linux nodes.....	354
IBM Spectrum Scale packaging overview.....	355
Preparing to install the IBM Spectrum Scale software on Linux nodes.....	355
Accepting the electronic license agreement on Linux nodes.....	355
Extracting the IBM Spectrum Scale software on Linux nodes.....	355
Verifying signature of IBM Spectrum Scale packages.....	357
Extracting IBM Spectrum Scale patches (update SLES and Red Hat Enterprise Linux RPMs or Ubuntu Linux packages).....	358
Installing the IBM Spectrum Scale man pages on Linux nodes.....	358
For Linux on Z: Changing the kernel settings.....	358
Manually installing the IBM Spectrum Scale software packages on Linux nodes.....	359
Building the GPFS portability layer on Linux nodes.....	364
Manually installing IBM Spectrum Scale and deploying protocols on Linux nodes.....	366
Verifying the IBM Spectrum Scale installation on Ubuntu Linux nodes.....	373
Verifying the IBM Spectrum Scale installation on SLES and Red Hat Enterprise Linux nodes.....	374
Manually installing IBM Spectrum Scale for object storage on Red Hat Enterprise Linux	374
Manually installing the performance monitoring tool.....	376
Manually installing IBM Spectrum Scale management GUI.....	379
Installing IBM Spectrum Scale on Linux nodes with the installation toolkit.....	386
Overview of the installation toolkit.....	386
Understanding the installation toolkit options.....	392
Limitations of the installation toolkit.....	394
Mixed operating system support with the installation toolkit.....	402
Preparing to use the installation toolkit.....	404
Using the installation toolkit to perform installation tasks: Explanations and examples.....	407
Populating cluster definition file with current cluster state using the installation toolkit.....	426
ESS awareness with the installation toolkit.....	429
Configuration of an IBM Spectrum Scale stretch cluster in an export services environment: a sample use case.....	431
Performing additional tasks using the installation toolkit.....	442
Protocol node IP further configuration.....	446
Object protocol further configuration.....	448
Enabling multi-region object deployment initially.....	448
Installing and using unified file and object access.....	449
Enabling unified file and object access after upgrading from IBM Spectrum Scale 4.2 or later.....	451
Chapter 5. Installing IBM Spectrum Scale on AIX nodes.....	453
Creating a file to ease the AIX installation process.....	453
Verifying the level of prerequisite software.....	453
Procedure for installing GPFS on AIX nodes.....	453
Accepting the electronic license agreement.....	454
Creating the GPFS directory.....	454
Creating the GPFS installation table of contents file.....	455
Installing the GPFS man pages.....	455
Installing GPFS over a network.....	455
Verifying the GPFS installation.....	455
Chapter 6. Installing IBM Spectrum Scale on Windows nodes.....	457
GPFS for Windows overview.....	457
GPFS support for Windows.....	458
GPFS limitations on Windows	459
File system name considerations.....	460
File name considerations.....	461
Case sensitivity.....	461

Antivirus software.....	462
Differences between GPFS and NTFS.....	463
Access control on GPFS file systems.....	464
Installing GPFS prerequisites.....	465
Configuring Windows.....	466
Installing Cygwin.....	466
Procedure for installing GPFS on Windows nodes.....	468
Running GPFS commands.....	469
Configuring a mixed Windows and UNIX (AIX or Linux) cluster.....	470
Configuring the Windows HPC server.....	474
Chapter 7. Installing Cloud services on IBM Spectrum Scale nodes.....	475
Creating a user-defined node class for Transparent cloud tiering or Cloud data sharing.....	475
Installation steps.....	476
Setting up a Cloud services cluster.....	477
Adding a Cloud services node to an existing Cloud services cluster.....	478
Chapter 8. Installing and configuring IBM Spectrum Scale management API.....	479
Chapter 9. Installing GPUDirect Storage for IBM Spectrum Scale.....	481
Chapter 10. Installation of Active File Management (AFM).....	483
Chapter 11. Installing AFM Disaster Recovery.....	485
Chapter 12. Installing call home.....	487
Chapter 13. Installing file audit logging.....	489
Requirements, limitations, and support for file audit logging.....	489
Requirements for using file audit logging with remotely mounted file systems.....	490
Chapter 14. Installing clustered watch folder.....	491
Requirements, limitations, and support for clustered watch folder.....	491
Requirements for using clustered watch folder with remotely mounted file systems.....	492
Manually installing clustered watch folder.....	492
Chapter 15. Steps to permanently uninstall IBM Spectrum Scale.....	495
Cleanup procedures required if reinstalling with the installation toolkit.....	496
Uninstalling the performance monitoring tool.....	502
Uninstalling the IBM Spectrum Scale management GUI.....	502
Removing nodes from management GUI-related node class.....	503
Permanently uninstall Cloud services and clean up the environment.....	503
Chapter 16. Upgrading.....	505
IBM Spectrum Scale supported upgrade paths.....	506
Online upgrade support for protocols and performance monitoring.....	507
Upgrading IBM Spectrum Scale nodes.....	508
Upgrading IBM Spectrum Scale non-protocol Linux nodes.....	509
Upgrading IBM Spectrum Scale protocol nodes.....	512
Upgrading GPUDirect Storage.....	517
Upgrading AFM and AFM DR.....	518
Using stop and start replication to upgrade AFM and AFM DR.....	519
Upgrading object packages.....	520
Upgrading SMB packages.....	521
Upgrading the SMB package after upgrading OS.....	523
Upgrading NFS packages.....	523

Upgrading call home.....	524
Call home configuration changes to be made while upgrading to IBM Spectrum Scale 5.0.x from IBM Spectrum Scale 4.2.1	524
Call home configuration changes to be made while upgrading to IBM Spectrum Scale 4.2.1 from IBM Spectrum Scale 4.2.0	525
Removing residual configuration files while upgrading to IBM Spectrum Scale 5.1.x.....	526
Manually upgrading the performance monitoring tool.....	526
Upgrading the performance monitoring packages after upgrading Ubuntu OS.....	527
Manually upgrading pmswift.....	527
Manually upgrading the IBM Spectrum Scale management GUI.....	529
Upgrading Cloud services.....	531
Upgrading to Transparent cloud tiering 1.1.2 from Transparent cloud tiering 1.1.0 or 1.1.1	531
Upgrading to Cloud services 1.1.2.1 from 1.1.2	532
Upgrading to Cloud services 1.1.3 from 1.1.2	532
Upgrading to Cloud services 1.1.3 from 1.1.0 and 1.1.1	533
Upgrading to Cloud services 1.1.4 from 1.1.2.x	534
Upgrading to Cloud services 1.1.4 from 1.1.3	535
Upgrading to Cloud services 1.1.5 from 1.1.4	536
Upgrading to Cloud services 1.1.6 from 1.1.5	537
Upgrading the Cloud services sensors	538
Upgrading to IBM Cloud Object Storage software level 3.7.2 and above.....	541
Upgrade paths and commands for file audit logging and clustered watch folder.....	542
Upgrading IBM Spectrum Scale components with the installation toolkit.....	543
Upgrade process flow.....	545
Performing online upgrade by using the installation toolkit.....	552
Performing offline upgrade or excluding nodes from upgrade by using installation toolkit.....	554
Upgrade rerun after an upgrade failure.....	560
Upgrading object packages to IBM Spectrum Scale 5.1.x or later by using the installation toolkit.....	561
Upgrading IBM Spectrum Scale on IBM Spectrum Archive Enterprise Edition (EE) nodes by using the installation toolkit.....	562
Protocol authentication configuration changes during upgrade.....	563
Changing the IBM Spectrum Scale product edition.....	566
Changing Express Edition to Standard Edition.....	571
Completing the upgrade to a new level of IBM Spectrum Scale.....	572
Reverting to the previous level of IBM Spectrum Scale.....	577
Reverting to a previous level of GPFS when you have <i>not</i> issued mmchconfig release=LATEST....	578
Reverting to a previous level of GPFS when you <i>have</i> issued mmchconfig release=LATEST.....	578
Coexistence considerations.....	579
Compatibility considerations.....	579
Considerations for IBM Spectrum Protect for Space Management.....	579
Applying maintenance to your IBM Spectrum Scale system.....	580
Guidance for upgrading the operating system on IBM Spectrum Scale nodes.....	580
Guidance for Red Hat Enterprise Linux 8.x on IBM Spectrum Scale nodes.....	583
Instructions for removing object protocol packages when upgrading protocol nodes to Red Hat Enterprise Linux 8.x.....	585
Considerations for upgrading from an operating system not supported in IBM Spectrum Scale 5.1.x.x.....	586
Servicing IBM Spectrum Scale protocol nodes.....	589
Offline upgrade with complete cluster shutdown.....	590
Accessibility features for IBM Spectrum Scale.....	593
Accessibility features.....	593
Keyboard navigation.....	593
IBM and accessibility.....	593
Notices.....	595
Trademarks.....	596

Terms and conditions for product documentation.....	596
Glossary.....	599
Index.....	607

Figures

1. A cluster with disks that are SAN-attached to all nodes.....	7
2. A cluster with some nodes connected to disks.....	7
3. A multicluster configuration.....	8
4. GPFS files have a typical UNIX structure.....	12
5. Sample of an AFM relationship	39
6. Global namespace implemented by using AFM.....	46
7. Read only mode.....	49
8. Single writer mode.....	49
9. Behaviors with local files.....	50
10. Independent writer mode.....	52
11. Sample setup of IBM Spectrum Protect for Space Management connected to home	85
12. IBM Spectrum Protect for Space Management connected to both home and cache.....	86
13. IW cache (Side A) to home (Side B).....	87
14. IW cache site (Side A) and home site (Side B).....	88
15. Asynchronous disaster recovery.....	97
16. AFM to cloud object storage.....	117
17. AFM to cloud object storage by using Azure Blob.....	119
18. IBM Spectrum Scale management API architecture.....	149
19. Option to download API description files in JSON format from the GUI node.....	158
20. REST API documentation opened in Swagger editor.....	159
21. Example for options that are available in the Swagger editor for each endpoint.....	160
22. Transparent cloud tiering and Cloud data sharing features.....	169
23. Exporting file system data to a cloud storage tier.....	172

24. Importing object storage data into the file system.....	173
25. WORM storage overview.....	174
26. Call home architecture.....	189
27. Guidance on which license to buy.....	216
28. GPFS configuration using node quorum	224
29. GPFS configuration using node quorum with tiebreaker disks.....	225
30. An example of a highly available SAN configuration for a GPFS file system.....	226
31. Configuration using GPFS replication for improved availability.....	227
32. Specifying a stanza for a thin-provisioned disk.....	242
33. Shared fast storage.....	288
34. Distributed fast storage.....	288
35. High-level overview of protocol user authentication.....	293
36. High-level flow of authentication for File protocols.....	295
37. IBM Spectrum Scale integration with internal Keystone server and external AD or LDAP authentication server.....	300
38. IBM Spectrum Scale for NFS architecture.....	302
39. IBM Spectrum Scale for Object Storage architecture.....	318
40. Synchronous mirroring with GPFS replication.....	431
41. Local Site CES Group.....	433

Tables

1. IBM Spectrum Scale library information units.....	xv
2. Conventions.....	xxxii
3. Supported devices on AIX.....	23
4. Supported devices on Linux.....	23
5. Supported devices on Windows.....	24
6. Comparison between NSD and NFS protocols.....	42
7. Conditions in which disabling AFM fileset fails, with corresponding messages	90
8. Supported function and supported OS/architecture.....	95
9. Supported function and supported OS/architecture.....	107
10. Features associated with IBM Spectrum Scale GUI pages.....	146
11. Getting information about filesets.....	152
12. Creating a snapshot.....	152
13. Response codes.....	155
14. Operations supported for resources and resource elements in API endpoints	161
15. File audit logging events.....	180
16. JSON attributes in file audit logging.....	182
17. File access events that are supported by clustered watch folder.....	185
18. RAS events that trigger data collection.....	192
19. Data collected and uploaded by call home.....	194
20. Features that IBM Spectrum Scale supports for deploying the OpenStack cloud storage.....	212
21. Features in IBM Spectrum Scale editions.....	214
22. GPFS cluster creation options.....	229
23. Thin provisioning storage solutions in IBM Spectrum Scale.....	241

24. Supported operating systems and device connections.....	246
25. Supported devices.....	246
26. File system creation options.....	252
27. Correspondence between mount options and the -S option in mmcrfs and mmchfs.....	261
28. Comparison of mmbackup and IBM Spectrum Protect Backup-Archive client backup commands....	273
29. General authentication support matrix.....	291
30. Tested NFS clients.....	303
31. Coherency option and description.....	313
32. OS support matrix.....	330
33. Recommended settings when you create a vault template on IBM Cloud Object Storage.....	333
34. Port requirements.....	335
35. workerThreads value per backend.....	342
36. GUI packages essential for each platform.....	380
37. Installation toolkit: List of features.....	388
38. spectrumscale command options for installing IBM Spectrum Scale and deploying protocols.....	393
39. Installation toolkit limitations.....	394
40. Operating systems supported with the installation toolkit in a mixed cluster.....	402
41. Validations by the installation toolkit in a mixed operating system cluster.....	403
42. Limitations of the config populate functionality.....	427
43. Generating short names for Windows.....	461
44. IBM Spectrum Scale supported online upgrade paths.....	506
45. Upgrade commands for file audit logging and clustered watch folder.....	542
46. Other options and parameters with mmchconfig release=LATEST.....	573
47. Upgrade paths if upgrade from OS not supported in IBM Spectrum Scale 5.1.x.x.....	588

About this information

This edition applies to IBM Spectrum Scale version 5.1.5 for AIX®, Linux®, and Windows.

IBM Spectrum 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 Spectrum Scale is running on a particular AIX node, enter:

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

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

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

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

To use these information units effectively, you must be familiar with IBM Spectrum 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 Spectrum Scale library information units		
Information unit	Type of information	Intended users
<i>IBM Spectrum 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 AFM• Planning for AFM DR• Planning for AFM to cloud object storage• Firewall recommendations	System administrators, analysts, installers, planners, and programmers of IBM Spectrum Scale clusters who are very experienced with the operating systems on which each IBM Spectrum Scale cluster is based
<i>IBM Spectrum Scale: Concepts, Planning, and Installation Guide</i>		

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

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

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

Information unit	Type of information	Intended users
<i>IBM Spectrum Scale: Concepts, Planning, and Installation Guide</i>	<ul style="list-style-type: none"> • Upgrading IBM Spectrum® Scale non-protocol Linux nodes • Upgrading IBM Spectrum Scale protocol nodes • Upgrading GPUDirect Storage • Upgrading AFM and AFM DR • Upgrading object packages • Upgrading SMB packages • Upgrading NFS packages • Upgrading call home • Manually upgrading the performance monitoring tool • Manually upgrading pmswift • Manually upgrading the IBM Spectrum 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 Spectrum Scale components with the installation toolkit • Protocol authentication configuration changes during upgrade • Changing the IBM Spectrum Scale product edition • Completing the upgrade to a new level of IBM Spectrum Scale • Reverting to the previous level of IBM Spectrum Scale 	System administrators, analysts, installers, planners, and programmers of IBM Spectrum Scale clusters who are very experienced with the operating systems on which each IBM Spectrum Scale cluster is based

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

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

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

Information unit	Type of information	Intended users
<i>IBM Spectrum 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 Spectrum 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 Spectrum Scale • Configuring file audit logging • Configuring clustered watch folder • 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 Spectrum Scale Cinder driver with Red Hat OpenStack Platform 16.1 • Configuring Multi-Rail over TCP (MROT) 	System administrators or programmers of IBM Spectrum Scale systems

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

Information unit	Type of information	Intended users
<i>IBM Spectrum 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 Spectrum Scale safeguarded copy • Verifying network operation with the mmnetverify command • Managing file systems • File system format changes between versions of IBM Spectrum Scale • Managing disks 	System administrators or programmers of IBM Spectrum Scale systems

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

Information unit	Type of information	Intended users
<i>IBM Spectrum Scale: Administration Guide</i>	<ul style="list-style-type: none"> • Managing protocol services • Managing protocol user authentication • Managing protocol data exports • 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 Spectrum 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 • Cloud services: Transparent cloud tiering and Cloud data sharing • Managing file audit logging • RDMA tuning • Configuring Mellanox Memory Translation Table (MTT) for GPFS RDMA VERBS Operation • Administering AFM • Administering AFM DR 	System administrators or programmers of IBM Spectrum Scale systems

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

Information unit	Type of information	Intended users
<i>IBM Spectrum 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 Spectrum Scale systems

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

Information unit	Type of information	Intended users
<i>IBM Spectrum 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 Spectrum Scale GUI • Monitoring system health by using the mmhealth command • 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 Spectrum 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 issues • 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 Spectrum Scale: Concepts, Planning, and Installation Guide</i></p>

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

Information unit	Type of information	Intended users
<i>IBM Spectrum 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 • Troubleshooting mmwatch • Maintenance procedures • Recovery procedures • Support for troubleshooting • References 	

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

Information unit	Type of information	Intended users
<i>IBM Spectrum Scale: Command and Programming Reference</i>	<p>This guide provides the following information:</p> <p>Command reference</p> <ul style="list-style-type: none"> • 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 Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XDSM standard

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

Information unit	Type of information	Intended users
<i>IBM Spectrum Scale: Command and Programming Reference</i>	<ul style="list-style-type: none"> • mmclone command • mmcloudgateway 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 • mmdelfs command • mmdelnod command • mmdelnodclass command • mmdelnod command • mmdelsnapshot command • mmdf command • mmdiag command • mmdsh command • mmeditac 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 Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XDSM standard

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

Information unit	Type of information	Intended users
<i>IBM Spectrum Scale: Command and Programming Reference</i>	<ul style="list-style-type: none"> • mmlinkfileset command • mmlsattr command • mmlscallback command • mmlscluster command • mmlsconfig command • mmlsdisk command • mmlsfileset command • mmlsfs command • mmlslicense command • mmlsmgr command • mmlsmount command • mmlsnodectrl command • mmlsnsd command • mmlspolicy command • mmlspool command • mmlsqos command • mmlsquota command • mmlssnapshot command • mmmigratefs command • mmmount command • mmnetverify command • mmnfs command • mmnsddiscover command • mmobj command • mmperfmon command • mmpmon command • mmprotocoltrace command • mmpsnap command • mmputacl command • mmqos command • mmquotaoff command • mmquotaon command • mmreclaimspace command • mmremotefluster command • mmremotefs command • mmrepquota command • mmrestoreconfig command • mmstorefs command • mmrestripefile command 	<ul style="list-style-type: none"> • System administrators of IBM Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XDSM standard

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

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

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

Information unit	Type of information	Intended users
<i>IBM Spectrum Scale: Big Data and Analytics Guide</i>	<p>This guide provides the following information:</p> <p>Summary of changes</p> <p>Big data and analytics support</p> <p>Hadoop Scale Storage Architecture</p> <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 <p>IBM Spectrum Scale support for Hadoop</p> <ul style="list-style-type: none"> • HDFS transparency overview • Supported IBM Spectrum 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 <p>IBM Spectrum Scale Hadoop performance tuning guide</p> <ul style="list-style-type: none"> • Overview • Performance overview • Hadoop Performance Planning over IBM Spectrum Scale • Performance guide 	<ul style="list-style-type: none"> • System administrators of IBM Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XDSE standard

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

Information unit	Type of information	Intended users
<i>IBM Spectrum Scale: Big Data and Analytics Guide</i>	<p>Cloudera Data Platform (CDP) Private Cloud Base</p> <ul style="list-style-type: none"> • Overview • Planning • Installing • Configuring • Administering • Upgrading • Limitations • Problem determination 	<ul style="list-style-type: none"> • System administrators of IBM Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XD SM standard
<i>IBM Spectrum Scale: Big Data and Analytics Guide</i>	<p>Cloudera HDP 3.X</p> <ul style="list-style-type: none"> • Planning • Installation • Upgrading and uninstallation • Configuration • Administration • Limitations • Problem determination <p>Open Source Apache Hadoop</p> <ul style="list-style-type: none"> • Open Source Apache Hadoop without CES HDFS • Open Source Apache Hadoop with CES HDFS <p>Cloudera HDP 2.6</p> <ul style="list-style-type: none"> • Planning • Installation • Upgrading software stack • Configuration • Administration • Troubleshooting • Limitations • FAQ 	<ul style="list-style-type: none"> • System administrators of IBM Spectrum Scale systems • Application programmers who are experienced with IBM Spectrum Scale systems and familiar with the terminology and concepts in the XD SM standard

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

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

Prerequisite and related information

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

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

Conventions used in this information

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

Note: Users of IBM Spectrum 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	<p>Bold words or characters represent system elements that you must use literally, such as commands, flags, values, and selected menu options.</p> <p>Depending on the context, bold typeface sometimes represents path names, directories, or file names.</p>
<u>bold underlined</u>	<u>bold underlined</u> keywords are defaults. These take effect if you do not specify a different keyword.
constant width	<p>Examples and information that the system displays appear in constant-width typeface.</p> <p>Depending on the context, constant-width typeface sometimes represents path names, directories, or file names.</p>
<i>italic</i>	<p><i>Italic</i> words or characters represent variable values that you must supply.</p> <p><i>Italics</i> are also used for information unit titles, for the first use of a glossary term, and for general emphasis in text.</p>
<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> .
\	<p>In command examples, a backslash indicates that the command or coding example continues on the next line. For example:</p> <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>.
item...	Ellipses indicate that you can repeat the preceding item one or more times.
	<p>In <i>synopsis</i> statements, vertical lines separate a list of choices. In other words, a vertical line means <i>Or</i>.</p> <p>In the left margin of the document, vertical lines indicate technical changes to the information.</p>

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 Spectrum 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 Spectrum 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 Spectrum Scale licensed program and the IBM Spectrum 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 Spectrum Scale 5.1.5.1 as updated, October 2022

This release of the IBM Spectrum Scale licensed program and the IBM Spectrum 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](#)
- [Documentation changes](#)

AFM and AFM DR-related changes

Introduced AFM IPv6 Support. For more information, see *AFM IPv6 Support* in the *IBM Spectrum Scale: Concepts, Planning, and Installation Guide*.

AFM to Cloud Object Storage

Introduced support of Seagate Lyve Cloud for AFM to cloud object storage. For more information, see *Introduction to AFM to cloud object storage* in the *IBM Spectrum Scale: Concepts, Planning, and Installation Guide*.

Big data and analytics changes

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

IBM Spectrum Scale Container Storage Interface driver changes

For information on changes in the IBM Spectrum Scale Container Storage Interface driver, see [IBM Spectrum Scale Container Storage Interface driver - Summary of changes](#).

IBM Spectrum Scale Erasure Code Edition changes

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

File system core improvements

Control fileset access for remote clusters

The list of allowed filesets for remote clusters can now be changed and be effective without a need to remount on remote nodes. For more information, see the topic *Fileset access control for remote clusters* in the *IBM Spectrum Scale: Administration Guide*.

New Safeguarded copy feature

The safeguarded copy feature is a mechanism to protect fileset and file system data from accidental or deliberate compromise. For more information, see the topic *Protecting file data: IBM Spectrum Scale safeguarded copy* in the *IBM Spectrum Scale: Administration Guide*.

New --expiration-time option for snapshot retention

The --expiration-time option specifies the expiration time of a snapshot for which a retention period is defined. It ensures that the snapshot cannot be deleted before its retention time expires. For more information, see the topic *mmcrsnapshot command* in the *IBM Spectrum Scale: Command and Programming Reference*.

New `--preview {policy|candidates}` option to execute the `mmbackup` command in the test only mode

This option allows the user to preview what would be processed by the `mmbackup` command without the backup cycles occurring. The option helps users to plan backups efficiently. The possible values are:

- `policy`: Generates the policy rules to be used in file selection and processing.
- `candidates`: Performs the policy scan and generates backup/expire candidate lists.

For more information, see the *mmbackup command* topic in the *IBM Spectrum Scale: Command and Programming Reference* guide.

New `mmrestrictedctl` command for performing specific system functions

The new `mmrestrictedctl` command is introduced to perform specific system functions. In IBM Spectrum Scale 5.1.5 it is used to delete a GPFS snapshot prior to the expiration date of the defined retention period. For more information, see the topic *mmrestrictedctl command* in the *IBM Spectrum Scale: Command and Programming Reference*.

Network resiliency enhanced with Multi-Rail over TCP (MROT) feature

The Multi-Rail over TCP (MROT) feature enables the concurrent use of multiple network interfaces and subnets. With MROT, the subnets attribute can be used to establish fault tolerance or automatic failover. All the IP addresses which are defined in the subnets attribute are used to establish connections with the nodes within the cluster. For more information, see the topic *Configuring Multi-Rail over TCP (MROT)* in the *IBM Spectrum Scale: Administration Guide*.

Define maximum number of filesets

The `maxFilesets` attribute helps define the maximum number of filesets that can be created in a file system. For more information, see the topic *mmchconfig command* in the *IBM Spectrum Scale: Command and Programming Reference*.

IBM Spectrum Scale supports accelerated writes with GPUDirect Storage (Tech Preview)

IBM Spectrum Scale supports accelerated writes with GPUDirect Storage as a Technical Preview feature. Accelerated GPUDirect Storage writes can be tested by the Licensee on non-production systems. The support of accelerated GPUDirect Storage reads and “writes in compatibility mode” remains unchanged including the use in production environments as in the previous release. For more information, see [GPUDirect Storage with accelerated writes support page](#) documentation.

IBM Spectrum Scale supports CipherTrust Manager 2.8

IBM Spectrum Scale supports CipherTrust Manager 2.8 for file system encryption. The following two configuration methods are supported depending on the certificates being used:

- Local certificate authority configuration

For more information, see the topic *Configuring encryption with the Thales CipherTrust Manager key server by using a local certificate authority* in the *IBM Spectrum Scale: Administration Guide*.

- External certificate authority configuration

For more information, see the topic *Configuring encryption with the Thales CipherTrust Manager key server by using an external certificate authority* in the *IBM Spectrum Scale: Administration Guide*.

Installation toolkit changes

- Ansible collection support in the toolkit.
- Precheck problem determination enhancement.
- Config populate enhancement.

Management API changes

The following endpoint is added:

- POST access

The following endpoints are modified:

- GET `Filesystems/{filesystemName}/filesets/{filesetName}/snapshots`

- GET filesystems/{filesystemName}/filesets/{filesetName}/snapshots/latest
- GET filesystems/{filesystemName}/filesets/{filesetName}/snapshots/snapshotName
- GET filesystems/{filesystemName}/snapshots
- GET filesystems/{filesystemName}/snapshots/snapshotName
- POST filesystems/{filesystemName}/filesets/{filesetName}/snapshots
- POST filesystems/{filesystemName}/snapshots
- POST filesystems/{filesystemName}/filesets/{filesetName}/directory/{path}
- PUT filesystems/{filesystemName}/unmount
- PUT filesystems/{filesystemName}/mount
- DELETE filesystems/{filesystemName}/filesets/{filesetName}/snapshots/{snapshotName}
- DELETE filesystems/{filesystemName}/snapshots/{snapshotName}

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

Management GUI changes

- The **About Us** page in the IBM Spectrum Scale management GUI displays the supported TLS versions.
- You can configure and edit S3 accounts, services and export through the **Data Access Service** panel.
- The IBM Spectrum Scale management GUI provides the facility to enable expiration of snapshots for which retention periods are defined.
- You can collect files from manual update (MU) filesets for a specific file system and upload in COS by using an AFM policies in the **Fileset tiering** panel.

SMB changes

Introduced **wide links** parameter that controls whether or not links in the UNIX file system might be followed by the server. For more information, see *MBps command* in the *IBM Spectrum Scale: Command and Programming Reference* guide.

Python-related changes

From IBM Spectrum Scale release 5.1.0, all Python code in the IBM Spectrum Scale product is converted to Python 3. The minimum supported Python version is 3.6.

For compatibility reasons on IBM Spectrum Scale 5.1.0.x and later on Red Hat Enterprise Linux 7.x (7.7 and later), a few Python 2 files are packaged and they might trigger dependency-related messages. In certain scenarios, Python 2.7 might also be required to be installed. Multiple versions of Python can co-exist on the same system. For more information, see the entry about **mmadquery** in *Guidance for Red Hat Enterprise Linux 8.x on IBM Spectrum Scale nodes* in *IBM Spectrum Scale: Concepts, Planning, and Installation Guide*.

The Python code in IBM Spectrum Scale 5.0.y or earlier continues to be in Python 2.

Tip:

- IBM Spectrum Scale 5.1.x.x uses Python 3 code, and it runs best with operating systems that also use Python 3 internally such as Red Hat Enterprise Linux 8.x, SLES 15, and Ubuntu 20.04.
- IBM Spectrum Scale 5.0.x.x uses Python 2 code, and it runs best with operating systems that also use Python 2 internally, such as Red Hat Enterprise Linux 7.x.

IBM Spectrum Scale 5.1.x supports Python 3.6 or later. It is recommended that Python 3.6 is installed through the OS package manager (For example, **yum install python3**). If you install Python 3.6 by other means, then unexpected results might occur, such as failure to install `gpfis.base` for prerequisite checks, and workarounds might be required.

System health changes

- Added a new sensor GPFSDiskCap with its sub-sensors in GPFS metrics as per the update in the perfmon config. For more information, see the *GPFS metrics in IBM Spectrum Scale: Problem Determination Guide*.
- New events added for the following:
 - GDS events
 - NVMeoF events

Commands, data types, and programming APIs

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

New commands

mmrestrictedctl

New structures

There are no new structures.

New subroutines

There are no new subroutines.

New user exits

There are no new user exits.

Changed commands

- **mmafmconfig**
- **mmafmctl**
- **mmafmcscctl**
- **mmauth**
- **mmbackup**
- **mmcallhome**
- **mmchconfig**
- **mmchfs**
- **mmcrcluster**
- **mmcrfs**
- **mmcrsnapshot**
- **mmfsckx**
- **mmhealth**
- **mmkeyserv**
- **mmlsfs**
- **mmlssnapshot**
- **mmsmb**
- **mmsdrrestore**
- **mmxcp**

Changed structures

There are no changed structures.

Changed subroutines

gpfs_fcctl()

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

6027-1312, 6027-3413, and 6027-4212

Changed messages

6027-2050 and 6027-3595

Deleted messages

6027-1922

Chapter 1. Introducing IBM Spectrum Scale

Overview of IBM Spectrum Scale

IBM Spectrum Scale is a clustered file system that provides concurrent access to a single file system or set of file systems from multiple nodes. The nodes can be SAN-attached, network attached, a mixture of SAN-attached and network attached, or in a shared nothing cluster configuration. This enables high-performance access to this common set of data to support a scale-out solution or to provide a high availability platform.

IBM Spectrum Scale has many features, beyond common data access, including data replication, policy-based storage management, and multisite operations. You can create a cluster of AIX nodes, Linux nodes, Windows server nodes, or a mix of all three. IBM Spectrum Scale can run on virtualized instances that provides common data access in environments, and uses logical partitioning or other hypervisors. Multiple IBM Spectrum Scale clusters can share data within a location or across wide area network (WAN) connections.

Strengths of IBM Spectrum Scale

IBM Spectrum Scale provides a global namespace, shared file system access among IBM Spectrum Scale clusters, simultaneous file access from multiple nodes, high recoverability and data availability through replication, the ability to make changes while a file system is mounted, and simplified administration even in large environments.

Shared file system access among IBM Spectrum Scale clusters

IBM Spectrum Scale allows you to share data between separate clusters within a location or across a WAN.

IBM Spectrum Scale clusters are independently managed, but IBM Spectrum Scale also shares data access through remote cluster mounts. This is known as a multicluster environment. When multiple clusters are configured to access the same IBM Spectrum Scale file system, IBM Global Security Kit (GSKit) is used to authenticate and check authorization for all network connections.

With GSKit, all messages within and across clusters are authenticated. You can also configure a cipherList to cause messages to be encrypted for transmissions.

The multicluster environment has the following features:

- The cluster that is hosting the file system can specify different security levels for each cluster authorized to mount a particular file system.
- The local cluster can remain active while changing security keys. Periodic changing of keys is necessary for a variety of reasons:
 - The number of keys should remain small to facilitate good performance.
 - Key changes prevent use or continued use of compromised keys.
 - As a matter of policy, some institutions require security keys to be changed periodically.

IBM Spectrum Scale uses public key authentication in a manner similar to the host-based authentication mechanism of OpenSSH. Each cluster has a pair of these keys that identify the cluster. In addition, each cluster also has an `authorized_keys` list. Each line in the `authorized_keys` list contains the public key of one remote cluster and a list of the file systems that the cluster is authorized to mount. For details about multicluster (remote mount) file system access, see *Accessing a remote GPFS file system* in *IBM Spectrum Scale: Administration Guide*.

For more information, see *Active file management* in the *IBM Spectrum Scale: Administration Guide*.

Improved system performance

GPFS file systems can improve system performance in a number of ways.

- Allowing multiple processes or applications on all nodes in the cluster simultaneous access to the same files. That is, it allows concurrent reads and writes from multiple nodes.
- Increasing aggregate bandwidth of your file system by spreading reads and writes across multiple disks.
- Balancing the load evenly across all disks to maximize their combined throughput, eliminating storage hotspots.
- Supporting very large file and file system sizes.
- Allowing concurrent reads and writes from multiple nodes.
- Provides sophisticated token management that handles fast and fine-grained access to cluster, file system, and file resources.
- Allowing for the specification of multiple networks for GPFS daemon communication and for GPFS administration command usage within your cluster.

Achieving high throughput to a single, large file that requires striping the data across multiple disks and disk controllers.

GPFS implements data striping in the file system instead of relying on data striping in a separate volume manager layer. As GPFS manages its own data striping, it enables the GPFS to achieve fault tolerance and balance load across adapters, storage controllers, and disks. Large files in GPFS are divided into equal sized blocks, and the consecutive blocks are placed on different disks in a round-robin manner.

GPFS automatically detects common data access patterns and automatically begins prefetching data accordingly. The prefetching and caching provide high throughput and fast response times. Some of the recognized I/O patterns include sequential, reverse sequential, and various forms of strided access patterns.

File consistency

IBM Spectrum Scale provides concurrent access to clients across the cluster by utilizing sophisticated token management. This provides concurrent and detailed access to IBM Spectrum Scale features, file systems, and file resources.

For more information, see [“GPFS architecture”](#) on page 8.

Increased data availability

GPFS provides multiple features that improve the reliability of your file system. This includes automatic features like file system logging and configurable features like intelligently mounting file systems on startup to providing tools for flexible synchronous replication.

GPFS allows you to organize your storage hardware into *failure groups*. A failure group is defined as a set of disks that share a common point of failure that could cause them all to become simultaneously unavailable. Failure groups are defined by the system administrator, so care needs to be taken when defining disks to ensure proper failure group isolation. When used in conjunction with the *replication* feature of GPFS, the creation of multiple failure groups provides for increased file availability should a group of disks fail. Replication in GPFS ensures that there is a copy of each block of replicated data and metadata on disks in different failure groups. In this case, should a set of disks become unavailable, GPFS fails over to the replicated copies in another failure group.

During configuration, you assign a replication factor to indicate the total number of copies of data and metadata you wish to store. Currently, the maximum replication factor is 3. Replication allows you to set different levels of protection for each file or one level for an entire file system. Since replication uses additional disk space and requires extra write time, you should consider the impact of replication on your application, especially if the replication is occurring over a WAN. To reduce the overhead involved with the replication of data, you can also choose to replicate only metadata as a means of providing additional file system protection. For more information on GPFS replication, see [“File system replication parameters”](#) on page 263.

GPFS is a logging file system. It creates separate logs for each file system. GPFS automatically replicates recovery logs if multiple failure groups are available. When used in conjunction with geographic based replication, disaster recovery abilities are provided. For more information on failure groups, see [“Network Shared Disk \(NSD\) creation considerations”](#) on page 239. For more information on disaster recovery with GPFS, see *Data Mirroring and Replication in IBM Spectrum Scale: Administration Guide*.

Once your file system is created, it can be configured to mount whenever the GPFS daemon is started. This feature assures that whenever the system and disks are up, the file system is available. When utilizing shared file system access among GPFS clusters to reduce overall GPFS control traffic, you might decide to mount the file system when it is first accessed. This is done either by running the `mmremotefs` command or the `mmchfs` command by using the `-A automount` option. GPFS mount traffic might be reduced by using automatic mounts instead of mounting at GPFS startup. Automatic mounts only produce additional control traffic at the point that the file system is first used by an application or user. Mounting at GPFS startup on the other hand produces additional control traffic at every GPFS startup. Thus, startup of hundreds of nodes at once is better served by using automatic mounts. However, when exporting the file system through Network File System (NFS) mounts, it might be useful to mount the file system when GPFS is started.

Enhanced system flexibility

With GPFS, your system resources are not frozen. You can add or delete disks while the file system is mounted.

When the time is favorable and system demand is low, you can rebalance the file system across all currently configured disks.

With the QoS capability, you can prevent I/O-intensive, long running administration commands from dominating file system performance and significantly delaying other tasks.

You can also add or delete nodes without having to stop and restart the GPFS daemon on all nodes.

Note: GPFS allows a large number of quorum nodes to facilitate maintaining quorum and continued cluster operation. GPFS also allows a tiebreaker disk configuration to further enhance cluster availability. For additional information, refer to [“Quorum”](#) on page 223.

If the physical connection to the disk is broken, GPFS dynamically switches disk access to the server nodes and continues to provide data through NSD server nodes. GPFS falls back to local disk access when it discovers that the path is repaired.

After GPFS is configured for your system, depending on your applications, hardware, and workload, you can re-configure GPFS to increase throughput. You can set up your GPFS environment for your current applications and users, secure in the knowledge that you can expand in the future without jeopardizing your data. GPFS capacity can grow as your hardware expands.

Simplified storage management

IBM Spectrum Scale can help you achieve information lifecycle management (ILM) through powerful policy-driven, automated tiered storage management.

IBM Spectrum Scale provides storage management based on the definition and use of:

- Storage pools
- Policies
- Filesets

Storage pools

A *storage pool* is a collection of disks or RAID configurations with similar properties that are managed together as a group. Storage pools provide a method to partition storage within a file system. While you plan how to configure your storage, consider factors 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 and impact of slower devices.

- Improved reliability by providing replication based on need and better failure containment.

Policies

Files are assigned to a storage pool based on defined *policies*. Policies provide for:

Placement policies

Placing files in a specific storage pool when the files are created.

File management policies

- Migrating files from one storage pool to another.
- Deleting files based on file characteristics.
- Changing the replication status of files.
- Snapshot metadata scans and file list creation.
- Compressing static files.

Filesets

Filesets provide a method for partitioning a file system and allow administrative operations at a finer granularity than the entire file system. For example, filesets allow you to:

- Define data block and inode quotas at the fileset level.
- Apply policy rules to specific fileset.
- Create snapshots at the fileset level.

For more information on storage pools, filesets, and policies, see *Information lifecycle management for IBM Spectrum Scale* in *IBM Spectrum Scale: Administration Guide*.

Simplified administration

GPFS offers many of the standard file system interfaces allowing most applications to execute without modification.

Operating system utilities complement the GPFS utilities, which means you can continue to also use the commands you have always used for ordinary file operations. For more information, see [“Considerations for GPFS applications” on page 346](#).

GPFS administration commands are similar in name and function to UNIX and Linux file system commands with one important difference: *the GPFS commands operate on multiple nodes*. A single GPFS command can perform an administration function across the entire cluster. For more information, see the individual commands in the *IBM Spectrum Scale: Command and Programming Reference*.

GPFS commands save configuration and file system information in one or more files. These are collectively known as GPFS cluster configuration data files. GPFS maintains the consistency of its configuration files across the cluster, and this provides accurate and consistent confirmation information. For more information, see [“Cluster configuration data files” on page 25](#).

Basic structure of IBM Spectrum Scale

IBM Spectrum Scale is a clustered file system that is defined over one or more nodes. On each node in the cluster, IBM Spectrum Scale consists of three basic components: administration commands, a kernel extension, and a multithreaded daemon.

For more information, see [“GPFS architecture” on page 8](#).

IBM Spectrum Scale administration commands

IBM Spectrum Scale administration commands are programs and scripts that control the configuration and operation of IBM Spectrum Scale. They typically begin with the letters “mm”, like in the **mmcrcluster** command.

By default, you can issue IBM Spectrum Scale administration commands from any node in a cluster. If the execution of an administration command requires tasks be performed on other nodes in the cluster, then the command automatically sends orders to the nodes to perform the tasks.

For this inter-node communication to succeed, one of the following setups is required:

- All the nodes in the cluster must be configured to allow passwordless remote shell communications.
- IBM Spectrum Scale sudo wrapper scripts must be configured on all the nodes of the cluster.

For more information about these two setups, see *Requirements for administering a GPFS file system* in the *IBM Spectrum Scale: Administration Guide*.

For descriptions of the commands, see *Command reference* in the *IBM Spectrum Scale: Command and Programming Reference*.

The GPFS kernel extension

The GPFS kernel extension provides the interfaces to the operating system vnode and virtual file system (VFS) layer to register GPFS as a native file system.

Structurally, applications make file system calls to the operating system, which presents them to the GPFS file system kernel extension. GPFS uses the standard mechanism for the operating system. In this way, GPFS appears to applications as just another file system. The GPFS kernel extension will either fulfill these requests by using resources which are already available in the system, or sends a message to the GPFS daemon to complete the request.

The GPFS daemon

The GPFS daemon performs all I/O operations and buffer management for GPFS. This includes read-ahead for sequential reads and write-behind for all writes that are not specified as synchronous. I/O operations are protected by GPFS token management, which ensures consistency of data across all nodes in the cluster.

The daemon is a multithreaded process with some threads dedicated to specific functions. Dedicated threads for services requiring priority attention are not used for or blocked by routine work. In addition to managing local I/O, the daemon also communicates with instances of the daemon on other nodes to coordinate configuration changes, recovery, and parallel updates of the same data structures. Specific functions that execute in the daemon include:

1. Allocation of disk space to new files and newly extended files, which is done in coordination with the [file system manager](#).
2. Management of directories including creation of new directories, insertion, and removal of entries into existing directories, and searching of directories that require I/O.
3. Allocation of appropriate locks to protect the integrity of data and metadata. Locks affect data that might be accessed from multiple nodes require interaction with the token management function.
4. Initiation of actual disk I/O on threads of the daemon.
5. Management of user security and quotas in conjunction with the file system manager.

The GPFS Network Shared Disk (NSD) component provides a method for cluster-wide disk naming and high-speed access to data for applications running on nodes that do not have direct access to the disks.

The NSDs in your cluster can be physically attached to all nodes or serve their data through an NSD server that provides a virtual connection. You are allowed to specify up to eight NSD servers for each NSD. If one server fails, the next server on the list takes control from the failed node.

For a given NSD, each of its NSD servers must have physical access to the same NSD. However, different servers can serve I/O to different non-intersecting sets of clients. The existing subnet functions in GPFS determine which NSD server should serve a particular GPFS client.

Note: GPFS assumes that nodes within a subnet are connected using high-speed networks. For more information on subnet configuration, see to [“Using public and private IP addresses for GPFS nodes”](#) on page 16.

GPFS determines whether a node has physical or virtual connectivity to an underlying NSD through a sequence of commands that are invoked from the GPFS daemon. This determination, which is called *NSD discovery*, occurs at initial GPFS startup and whenever a file system is mounted.

Note: To manually cause this discovery action, use the `mmnsddiscover` command. For more information, see *mmnsddiscover command* in *IBM Spectrum Scale: Command and Programming Reference*.

This is the default order of access used during NSD discovery:

1. Local block device interfaces for SAN, SCSI, IDE, or DASD disks
2. NSD servers

This order can be changed with the `useNSDserver` mount option.

You must define NSD servers for the disks. In a SAN configuration where NSD servers are defined, if the physical connection is broken, GPFS dynamically switches to the server nodes and continues to provide data. GPFS falls back to local disk access when the discovered path is repaired. This is the default behavior, and it can be changed with the `use NSD server file system` mount option.

For more information, see [“Disk considerations” on page 238](#) and [“NSD disk discovery” on page 23](#).

The GPFS open source portability layer

On Linux platforms, GPFS uses a loadable kernel module that enables the GPFS daemon to interact with the Linux kernel. Source code is provided for the portability layer so that the GPFS portability can be built and installed on a wide variety of Linux kernel versions and configuration.

When installing GPFS on Linux, you build a portability module based on your particular hardware platform and Linux distribution to enable communication between the Linux kernel and GPFS. For more information, see [“Building the GPFS portability layer on Linux nodes” on page 364](#).

IBM Spectrum Scale cluster configurations

An IBM Spectrum Scale cluster can be configured in a variety of ways. The cluster can be a heterogeneous mix of hardware platforms and operating systems.

IBM Spectrum Scale clusters can contain a mix of all supported node types including Linux, AIX, and Windows Server and these operating systems can run on various hardware platforms, such as IBM POWER®, x86-based servers, and IBM Z®. These nodes can all be attached to a common set of SAN storage or through a mix of SAN and network attached nodes. Nodes can all be in a single cluster, or data can be shared across multiple clusters. A cluster can be contained in a single data center or spread across geographical locations. To determine which cluster configuration is best for your application, start by determining the following:

- Application I/O performance and reliability requirements.
- Properties of the underlying storage hardware.
- Administration, security, and ownership considerations.

Understanding these requirements helps you determine which nodes require direct access to the disks and which nodes should access the disks over a network connection through an NSD server.

There are four basic IBM Spectrum Scale configurations:

- When all nodes are attached to a common set of Logical Unit Numbers (LUNs).
- When some nodes are NSD clients.
- When a cluster is spread across multiple sites.
- When data is shared between clusters.

All nodes attached to a common set of LUNs

In this type of configuration, all of the nodes in the cluster are connected to a common set of LUNs (for example, over a SAN). The following factors must be considered while defining this configuration:

- The maximum number of nodes accessing a LUN that you want to support.
- You cannot mix different operating systems with IBM Spectrum Scale to directly access the same set of LUNs on SAN.

For example, see [Figure 1 on page 7](#).

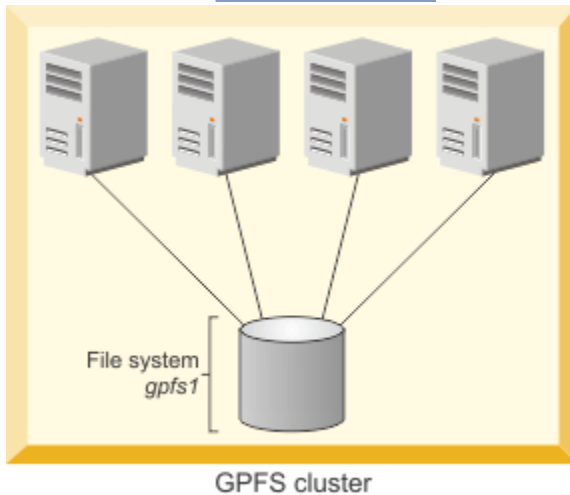


Figure 1. A cluster with disks that are SAN-attached to all nodes

Some nodes are NSD clients

In this type of configuration, only some nodes are connected to disks. Other nodes access the disks using the NSD path.

For an example, see [Figure 2 on page 7](#).

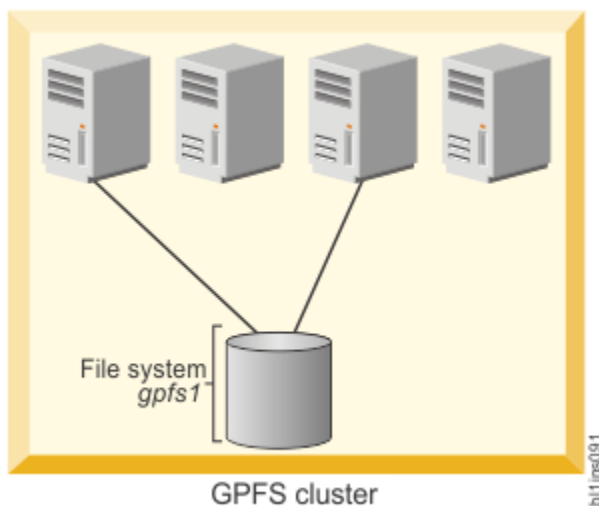


Figure 2. A cluster with some nodes connected to disks

IBM Spectrum Scale servers and clients

You can configure an IBM Spectrum Scale cluster in which some nodes have a direct attachment to the disks and others access the disks through other IBM Spectrum Scale nodes. This configuration is often used in large clusters or to provide a cost-effective and high-performance solution.

When an IBM Spectrum Scale node is providing access to a disk for another IBM Spectrum Scale node, the node that provides access is called an NSD server. The node that accesses the data through an NSD server is called an IBM Spectrum Scale client.

Sharing data across multiple IBM Spectrum Scale clusters

IBM Spectrum Scale allows you to share data across multiple IBM Spectrum Scale clusters. After a file system is mounted in another IBM Spectrum Scale cluster, all access to the data is the same as if you were in the host cluster. You can connect multiple clusters within the same data center or across

long distances over a WAN. In a multicluster configuration, each cluster can be placed in a separate administrative group simplifying administration or provide a common view of data across multiple organizations.

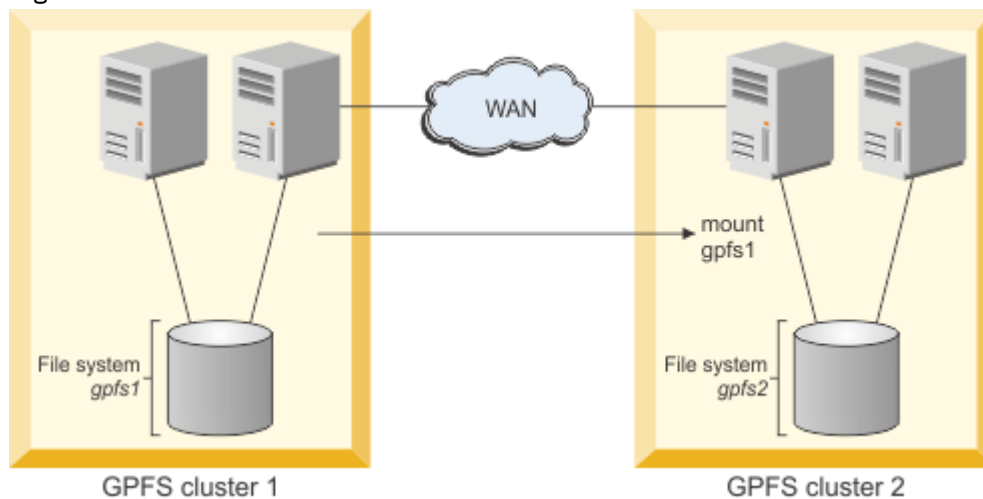


Figure 3. A multicluster configuration

Note: For more information, see *Accessing a remote GPFS file system in IBM Spectrum Scale: Administration Guide*.

GPFS architecture

Use this information to understand the architecture of GPFS.

Interaction between nodes at the file system level is limited to the locks and control flows that are required to maintain data and metadata integrity in the parallel environment.

A discussion of GPFS architecture includes:

- [“Special management functions” on page 8](#)
- [“Use of disk storage and file structure within a GPFS file system” on page 11](#)
- [“GPFS and memory” on page 14](#)
- [“GPFS and network communication” on page 15](#)
- [“Application and user interaction with GPFS” on page 18](#)
- [“NSD disk discovery” on page 23](#)
- [“Failure recovery processing” on page 24](#)
- [“Cluster configuration data files” on page 25](#)
- [“GPFS backup data” on page 26](#)

Special management functions

In general, GPFS performs the same functions on all nodes. It handles application requests on the node where the application exists. This provides maximum affinity of the data to the application.

In the following cases, one node provides a more global function affecting the operation of multiple nodes. These are nodes acting as:

1. [“The GPFS cluster manager” on page 9](#)
2. [“The file system manager” on page 9](#)
3. [“The metanode” on page 11](#)
4. [“CES node \(protocol node\)” on page 11](#)
5. [“AFM gateway node” on page 11](#)

Note:

- For quorum nodes, see “Quorum” on page 223.
- For Cloud services nodes, see *Designating the Cloud services nodes* in the *IBM Spectrum Scale: Administration Guide*.

The GPFS cluster manager

There is one GPFS cluster per manager, which is selected from a set of quorum nodes that are designated for the cluster.

See “Quorum” on page 223 for more information.

The cluster manager performs the following tasks:

- Monitors disk leases.
 - Detects failures and manages recovery from node failure within the cluster.
- The cluster manager determines whether a quorum of nodes exists to allow the GPFS daemon to start and for file system usage to continue.
- Distributes certain configuration changes that must be known to nodes in remote clusters.
 - Selects the file system manager node.

The cluster manager prevents multiple nodes from assuming the role of file system manager to avoid data corruption. The token management service resides on the file system manager node and any other nodes you specify. For more information, see *Using multiple token servers* in *IBM Spectrum Scale: Administration Guide*.

- Handles UID mapping requests from remote cluster nodes.
- Aggregates health information from all nodes in the cluster.

To identify the cluster manager, issue the **mmfsmgr -c** command. For more information, see *mmfsmgr command* in *IBM Spectrum Scale: Command and Programming Reference*.

To change the cluster manager, issue the **mmchmgr -c** command. For more information, see *mmchmgr command* in *IBM Spectrum Scale: Command and Programming Reference*.

The file system manager

There is one file system manager per file system, which handles all of the nodes by using the file system.

The services provided by the file system manager include:

1. File system configuration

Processes the following file system changes:

- Adding disks
- Changing disk availability
- Repairing the file system

Mount and unmount processing is performed on both the file system manager and the node that are requesting the service.

2. Management of disk space allocation

Controls which regions of disks are allocated to each node, allowing effective parallel allocation of space.

3. Token management

The file system manager node might also perform the duties of the token manager server. If you have explicitly designated some of the nodes in your cluster as file system manager nodes, then the token server load is distributed among all of the designated manager nodes. For more information, see *Using multiple token servers* in *IBM Spectrum Scale: Administration Guide*.

The token management server coordinates access to files on shared disks by granting tokens that convey the right to read or write the data or metadata of a file. This service ensures the consistency of the file system data and metadata when different nodes access the same file. The status of each token is held in two places:

- a. On the token management server
- b. On the token management client holding the token

The first time a node accesses a file it must send a request to the token management server to obtain a corresponding read or write token. After having been granted the token, a node might continue to read or write to the file without requiring additional interaction with the token management server. This continues until an application on another node attempts to read or write to the same region in the file.

The normal flow for a token is:

- A message to the token management server.

The token management server then either returns a granted token or a list of the nodes that are holding conflicting tokens.

- The token management function at the requesting node then has the responsibility to communicate with all nodes holding a conflicting token and get them to relinquish the token.

This relieves the token server of having to deal with all nodes holding conflicting tokens. In order for a node to relinquish a token, the daemon must give it up. First, the daemon must release any locks that are held using this token, which might involve waiting for I/O to complete.

4. Quota management

In a quota-enabled file system, the file system manager node automatically assumes quota management responsibilities whenever the GPFS file system is mounted. Quota management involves:

- Allocating disk blocks to nodes that are writing to the file system
- Comparing the allocated space to the quota limits at regular intervals

Notes:

- a. To reduce the number of space requests from nodes writing to the file system, the quota manager allocates more disk blocks than requested (see [“Enabling quotas” on page 265](#)). This allows nodes to write to the file system without having to go to the quota manager and check quota limits each time they write to the file system.
- b. Quota-enabled file systems with more than 100,000 users or groups must avoid designating nodes as manager. This situation is avoided because nodes might have low memory or they might get heavily loaded due to high-memory demands for quota manager operations.

The file system manager is selected by the cluster manager. If a file system manager fails for any reason, then a new file system manager is selected by the cluster manager. During this transition, all functions continue without disruption, except for the time required to accomplish the takeover.

Depending on the application workload, the memory and CPU requirements for the services that are provided by the file system manager can make it undesirable to run a resource intensive application on the same node as the file system manager. GPFS allows you to control the pool of nodes from which the file system manager is selected by using the following commands:

- The `mmcrcluster` command when creating your cluster.
- The `mmaddnode` command when adding nodes to your cluster.
- The `mmchnode` command to change a node's designation at any time.

These preferences are honored except in certain failure situations where multiple failures occur. For more information, see *Multiple file system manager failures in IBM Spectrum Scale: Problem Determination Guide*. You might list which node is currently assigned as the file system manager by issuing the `mm1smgr` command or change which node is assigned to this task by using the `mmchmgr` command.

The metanode

There is one metanode per open file. A file refers to a file system object that also includes a directory. The metanode is responsible for maintaining file metadata integrity.

In almost all cases, the node that has had the file open for the longest continuous period is the metanode. All nodes accessing a file can read and write data directly, but updates to metadata are written only by the metanode. The metanode for each file can move to any node to meet application requirements, except when multiple remote clusters access the same file. In such a scenario, the metanode is placed on a home cluster node. The "home cluster" is also known as the "owning cluster".

CES node (protocol node)

Only nodes that are designated as CES nodes can serve integrated protocol function.

Nodes in the cluster can be designated to be CES nodes by using the `mmchnode --ces-enable Node` command. Each CES node serves each of the protocols (NFS, SMB, Object, Block, HDFS namenode) that are enabled. CES IP addresses that are assigned for protocol serving can failover to any of CES nodes that are up based on the configured failback policy. CES functionality can be designated only on nodes that are running on supported operating systems, and all the CES nodes must have the same platform (either all Intel or all POWER Big Endian).

For information about supported operating systems and their required minimum kernel levels, see [IBM Spectrum Scale FAQ in IBM Documentation](#).

For more information about Cluster Export Services, see *Implementing Cluster Export Services* and *Configuring the CES and protocol configuration* in *IBM Spectrum Scale: Administration Guide*.

Related concepts

[“Protocols support overview: Integration of protocol access methods with GPFS” on page 28](#)

IBM Spectrum Scale provides extra protocol access methods. Providing these additional file and object access methods and integrating them with GPFS offers several benefits. It enables users to consolidate various sources of data efficiently in one global namespace. It provides a unified data management solution and enables not only efficient space utilization but also avoids making unnecessary data moves just because access methods might be different.

AFM gateway node

Each cache fileset in a cluster is served by one of the nodes that is designated as the gateway in the cluster.

The gateway node that is mapped to a fileset is called the primary gateway of the fileset. The primary gateway acts as the owner of the fileset and communicates with the home cluster. While you are using parallel data transfers, the primary gateway might also communicate with other participating gateway nodes for data transfer to home clusters.

All other nodes in the cluster, including other gateways, become the application nodes of the fileset. Therefore, any node in the cache cluster can function as a gateway node and an application node for different filesets based on configuration of the node.

Use of disk storage and file structure within a GPFS file system

A file system (or stripe group) consists of a set of disks that store file data, file metadata, and supporting entities, such as quota files and recovery logs.

When a disk is assigned to a file system, a file system descriptor is written on each disk. The *file system descriptor* is written at a fixed position on each of the disks in the file system and is used by GPFS to identify this disk and its place in a file system. The file system descriptor contains file system specifications and information about the state of the file system.

Within each file system, files are written to disk as in other UNIX file systems, using inodes, indirect blocks, and data blocks. Inodes and indirect blocks are considered *metadata*, as distinguished from data,

or actual file content. You can control which disks GPFS uses for storing metadata when you create the file system with the `mmcrfs` command or when you modify the file system with the `mmchdisk` command.

The metadata for each file is stored in the inode and contains information such as file size and time of last modification. The inode also sets aside space to track the location of the data of the file. On file systems that are created in IBM Spectrum Scale, if the file is small enough that its data can fit within this space, the data can be stored in the inode itself. This method is called data-in-inode and improves the performance and space utilization of workloads that use many small files. Otherwise, the data of the file must be placed in data blocks, and the inode is used to find the location of these blocks. The location-tracking space of the inode is then used to store the addresses of these data blocks. If the file is large enough, the addresses of all of its data blocks cannot be stored in the inode itself, and the inode points instead to one or more levels of indirect blocks. These trees of additional metadata space for a file can hold all of the data block addresses for large files. The number of levels that are required to store the addresses of the data block is referred to as the *indirection level* of the file.

To summarize, on file systems that are created in IBM Spectrum Scale, a file typically starts out with data-in-inode. When it outgrows this stage, the inode stores direct pointers to data blocks; this arrangement is considered a zero level of indirection. When more data blocks are needed, the indirection level is increased by adding an indirect block and moving the direct pointers there; the inode then points to this indirect block. Subsequent levels of indirect blocks are added as the file grows. The dynamic nature of the indirect block structure allows file sizes to grow up to the file system size.

For security reasons, encrypted files skip the data-in-inode stage. They always begin at indirection level zero.

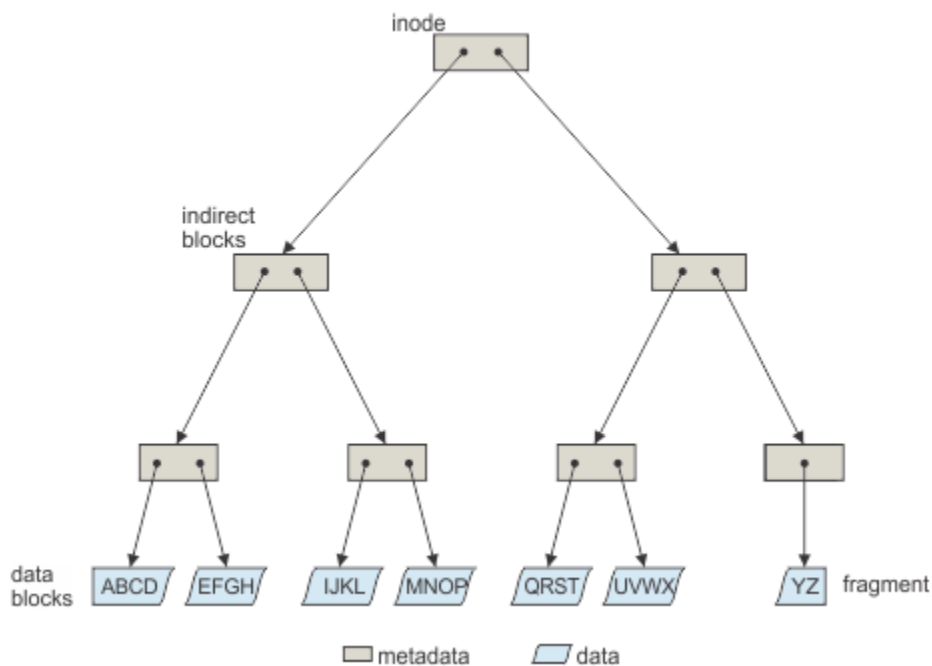


Figure 4. GPFS files have a typical UNIX structure

File system limitations:

1. The maximum number of mounted file systems within a GPFS cluster is 256.
2. The supported file system size depends on the version of GPFS that is installed.
3. The maximum number of files within a file system cannot exceed the architectural limit.

For the latest information on these file system limitations, see the [IBM Spectrum Scale FAQ in IBM Documentation](#).

GPFS uses the file system descriptor to find all of the disks that make up the file system's stripe group, including their size and order. Once the file system descriptor is processed, it is possible to address any block in the file system. In particular, it is possible to find the first inode, which describes the *inode file*,

and a small number of inodes that contain the rest of the file system information. The inode file is a collection of fixed-length records that represent a single file, directory, or link. The unit of locking is the single inode. Specifically, there are fixed inodes within the inode file for the following components:

- Root directory of the file system.
- *Block allocation map*, which is a collection of bits that represent the availability of disk space within the disks of the file system. One unit in the allocation map represents a subblock. A subblock is the smallest unit of contiguous disk space that can be allocated to a file. Block size, subblock size, and the number of subblocks per block are set when the file system is created and cannot be changed afterward. For more information, see the *mmcrfs* command in the *IBM Spectrum Scale: Command and Programming Reference*. The allocation map is broken into regions that reside on disk sector boundaries. The number of regions is set at file system creation time by the parameter that specifies how many nodes access this file system. The regions are separately locked and as a result, different nodes can be allocating or deallocating space that is represented by different regions independently and concurrently.
- *Inode allocation map*, which represents the availability of inodes within the inode file. The *Inode allocation map* is located in the *inode allocation file*, and represents all the files, directories, and links that can be created. The *mmchfs* command can be used to change the maximum number of files that can be created in the file system up to the architectural limit.

The data contents of each of these files are taken from the data space on the disks. These files are considered metadata and are allocated only on disks where metadata is allowed. For more information, see the *mmcrfs* command in the *IBM Spectrum Scale: Command and Programming Reference*.

Quota files

For each file system, IBM Spectrum Scale maintains quota information in internal quota files: a quota file for users, a quota file for groups, and a quota file for filesets.

Quota files are created when quotas are enabled by the `-Q yes` option, which can be either set in the **mmcrfs** command or in the **mmchfs** command. Quota files are maintained until quotas are disabled by the **mmchfs -Q no** command. To determine whether quotas are enabled on a file system, issue the following command:

```
mmfsfs Device -Q
```

where *Device* is the file system.

Quota files are internal product files and are not available for access by users. Quota file information is backed up by the **mmbackupconfig** command and is restored by the **mmrestoreconfig** command.

The quota files contain the following information:

- The quota file for users contains the block limits, file limits, actual usage, and grace period for each user.
- The quota file for groups contains the block limits, file limits, actual usage, and grace period for each group.
- The quota file for filesets contains the block limits, file limits, actual usage, and grace period for each fileset.

GPFS recovery logs

GPFS recovery logs are created at file system creation. Additional recovery logs are automatically created as needed. The file system manager assigns a recovery log to each node that accesses the file system.

GPFS maintains the atomicity of the on-disk structures of a file through a combination of rigid sequencing of operations and logging. The data structures maintained are the inode, indirect blocks, the allocation map, and the data blocks. Data blocks are written to disk before any control structure that references the data is written to disk. This ensures that the previous contents of a data block can never be seen in a new file. Allocation blocks, inodes, and indirect blocks are written and logged in such a way that there can never be a pointer to a block marked unallocated that is not recoverable from a log.

Recovery logs are replicated only if default metadata replication is turned on (`-m 2`) or if explicitly enabled for logs (`--log-replicas 2`). You can check to see if log replication is enabled for a file system by using the `mmfsfs` command and looking at the value of the `-m` and `--log-replicas` parameters. If both are set, the `--log-replicas` value takes precedence over the `-m` value for log replication.

There are certain failure cases where blocks are marked allocated but not yet assigned to a file, and these can be recovered by running the `mmfsck` command in online or offline mode. Log recovery is run as part of:

1. The recovery of a node failure affecting the objects that the failed node might have had locked.
2. A mount after the file system has been unmounted everywhere.

GPFS and memory

GPFS uses three areas of memory: memory allocated from the kernel heap, memory allocated within the daemon segment, and shared segments accessed from both the daemon and the kernel.

Memory allocated from the kernel heap

GPFS uses kernel memory for control structures such as vnodes and related structures that establish the necessary relationship with the operating system.

Memory allocated within the daemon segment

GPFS uses daemon segment memory for file system manager functions. Because of that, the file system manager node requires more daemon memory since token states for the entire file system are initially stored there. File system manager functions requiring daemon memory include:

- Structures that persist for the execution of a command
- Structures that persist for I/O operations
- States related to other nodes

The file system manager is a token manager, and other nodes might assume token management responsibilities. Therefore, any manager node can consume additional memory for token management. For more information, see *Using multiple token servers* in *IBM Spectrum Scale: Administration Guide*.

Shared segments accessed from both the daemon and the kernel

Shared segments consist of both pinned and unpinned memory that is allocated at daemon startup. The initial values are the system defaults. However, you can change these values later by using the `mmchconfig` command. See [“Cluster configuration file” on page 237](#).

The pinned memory is called the *pagepool* and is configured by setting the `pagepool` cluster configuration attribute. This pinned area of memory is used for storing file data and for optimizing the performance of various data access patterns. In a non-pinned area of the shared segment, GPFS keeps information about open and recently opened files. This information is held in two forms:

1. A full inode cache
2. A stat cache

Pinned memory

GPFS uses pinned memory (also called page pool memory) for storing file data and metadata in support of I/O operations.

With some access patterns, increasing the amount of page pool memory can increase I/O performance. Increased page pool memory can be useful in the following cases:

- There are frequent writes that can be overlapped with application execution.
- There is frequent reuse of file data that can fit in the page pool.
- The I/O pattern contains various sequential reads large enough that the prefetching of data improves performance.

Pinned memory regions cannot be swapped out to disk, which means that GPFS always consumes at least the value of the `pagepool` attribute in system memory. So, consider the memory requirements of GPFS and other applications running on the node when determining a value for the `pagepool` attribute.

Non-pinned memory

There are two levels of cache used to store file metadata.

Inode cache

The inode cache contains copies of inodes for open files and for some recently used files that are no longer open. The **maxFilesToCache** parameter controls the number of inodes cached by GPFS. Every open file on a node consumes a space in the inode cache. Additional space in the inode cache is used to store the inodes for recently used files in case another application needs that data.

The number of open files can exceed the value defined by the **maxFilesToCache** parameter to enable applications to operate. However, when the **maxFilesToCache** number is exceeded, there is no more caching of recently open files, and only open file inode data is kept in the cache.

Stat cache

The stat cache contains enough information to respond to inquiries about the file and open it, but not enough information to read from it or write to it. There is sufficient data from the inode in the stat cache to respond to a `stat()` call (for example, when issuing the `ls -l` command on a UNIX node). A stat cache entry consumes significantly less memory than a full inode. Stat cache entries are kept for the following:

- Recently accessed files
- Directories recently accessed by a number of `stat()` calls

To set the value of the stat cache, use the **maxStatCache** attribute of the **mmchconfig** command.

Notes:

1. GPFS prefetches data for stat cache entries if a pattern of use indicates this is productive (for example, if a number of `ls -l` commands issued for a large directory).
2. In versions of IBM Spectrum 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 Spectrum Scale: Command and Programming Reference*. The size of the GPFS shared segment can limit the maximum setting of **maxStatCache**.
3. Each entry in the inode cache and the stat cache requires appropriate tokens:
 - a. To ensure the cached information remains correct
 - b. For the storage of tokens on the file system manager node
4. Depending on the usage pattern, system performance might degrade when an information update requires revoking a token. This happens when two or more nodes share the same information and the most recent information is moved to a different location. When the current node needs to access the updated information, the token manager must revoke the token from the current node before that node can access the information in the new location.

GPFS and network communication

Within the GPFS cluster, you can specify different networks for GPFS daemon communication and for GPFS command usage.

You can select the different networks by issuing **mmaddnode**, **mmchnode**, and **mmcrcluster** commands. In these commands, the node descriptor allows you to specify separate node interfaces for those functions on each node. The correct operation of GPFS is directly dependent upon these selections.

GPFS might not work properly if there is a firewall enabled on the nodes within the cluster. To ensure proper operation, you must either configure the firewall to allow the appropriate ports or disable the

firewall. For more information, see *GPFS port usage* in *IBM Spectrum Scale: Administration Guide* and *mmnetverify command* in *IBM Spectrum Scale: Command and Programming Reference*.

GPFS daemon communication

In a cluster environment, the GPFS daemon depends on the correct operation of TCP/IP.

These dependencies exist because:

- The communication path between nodes must be built at the first attempt to communicate.
- Each node in the cluster is required to communicate with the cluster manager and the file system manager during startup and mount processing.
- Once a connection is established, it must remain active until the GPFS daemon is shut down on the nodes.

Note: Establishing other communication paths depends upon application usage among nodes.

The daemon also uses sockets to communicate with other instances of the file system on other nodes. Specifically, the daemon on each node communicates with the file system manager for allocation of logs, allocation segments, and quotas, as well as for various recovery and configuration flows. GPFS requires an active internode communications path between all nodes in a cluster for locking, metadata coordination, administration commands, and other internal functions. The existence of this path is necessary for the correct operation of GPFS. The instance of the GPFS daemon on a node goes down if it senses that this communication is not available to it. If communication is not available to another node, then one of the two nodes exit GPFS.

Using public and private IP addresses for GPFS nodes

GPFS permits the system administrator to set up a cluster such that both public and private IP addresses are in use. For example, if a cluster has an internal network connecting some of its nodes, it is advantageous to use private IP addresses to communicate on this network, and public IP addresses to communicate to resources outside of this network.

Public IP addresses are those that can be used to access the node from any other location for which a connection exists. Private IP addresses might be used only for communications between nodes directly connected to each other with a communications adapter. Private IP addresses are assigned to each node at hardware setup time, and must be in a specific address range (IP addresses on a 10.0.0.0, 172.16.0.0, or 192.168.0.0 subnet). For more information on private IP addresses, refer to [RFC 1597 - Address Allocation for Private Internets \(www.ip-doc.com/rfc/rfc1597\)](http://www.ip-doc.com/rfc/rfc1597).

The subnets operand on the `mmchconfig` command specifies an ordered list of subnets available to GPFS for private TCP/IP communications. Each listed subnet might have a list of cluster names (allowing shell-style wild cards) that specifies other GPFS clusters that have direct access to the same subnet.

When the GPFS daemon starts on a node, it obtains a list of its own IP addresses and associated subnet masks from its local IP configuration. For each IP address, GPFS checks whether that address is on one of the subnets specified on the subnets configuration parameter. It records the list of its matching IP addresses and subnet masks, and then listens for connections on any of these addresses. If any IP address for the node (specified when the cluster was created or when the node was added to the cluster), is not specified with the subnets configuration parameter, GPFS automatically adds it to the end of the node's IP address list.

Therefore, when you use public IP addresses for a node, there is no need to explicitly list the public IP subnet with the subnets configuration parameter. For example, the normal way to configure a system would be to use host names that resolve to the external Ethernet IP address in the `mmcrcluster` command, and then, if the system administrator wants GPFS to use the High Performance Switch within the cluster, add one subnets configuration parameter for the HPS subnet. It is acceptable to add two subnets configuration parameters, one for the HPS and one for the external Ethernet, making sure that they are in that order. In this case, it does not matter which of each node's two addresses was specified when the cluster was created or when the node was added to the cluster. For example, to add remote access to an existing cluster that was using only switch addresses, it is sufficient to add two subnets configuration parameters.

When a node joins a cluster (its own cluster on startup, or another cluster when mounting a file system owned by another cluster), the node sends its list of IP addresses (ordered according to the order of subnets configuration parameters) to the cluster manager node, which forwards the list to all other nodes as part of the join protocol. No other additional information needs to be propagated.

When a node attempts to establish a connection to another node, GPFS determines the destination IP address to use according to this procedure:

1. For each of its own IP addresses, it searches the other node's list of IP addresses for an address that is on the same subnet.
 - For normal public IP addresses this is done by comparing IP address values ANDed with the node's subnet mask for its IP address.
 - For private IP addresses GPFS assumes that two IP addresses are on the same subnet only if the two nodes are within the same cluster, or if the other node is in one of the clusters that are explicitly listed in the subnets configuration parameter.
2. If the two nodes have more than one IP address pair on a common subnet, GPFS uses the first one found according to the order of subnets specified in the initiating node's configuration parameters.
3. If a given pair of nodes do not share a common subnet defined by the subnets configuration parameter, IBM Spectrum Scale uses the network based on the daemon address of the node, which is automatically added as the last entry in each node's network address list.

Note: When you use subnets, both the interface corresponding to the daemon address and the interface that matches the subnet settings must be operational.

For more information and an example, see *Using remote access with public and private IP addresses* in *IBM Spectrum Scale: Administration Guide*.

Network communication and GPFS administration commands

Socket communications are used to process GPFS administration commands. Depending on the nature of the command, GPFS might process commands either on the node by issuing the command or on the file system manager. The actual command processor merely assembles the input parameters and sends them along to the daemon on the local node by using a socket.

Some GPFS commands permit you to specify a separate administrative network name. You make this specification by using the AdminNodeName field of the node descriptor. For additional information, see *IBM Spectrum Scale: Command and Programming Reference* for descriptions of these commands:

- `mmaddnode`
- `mmchnode`
- `mmcrcluster`

If the command changes the state of a file system or its configuration, the command is processed at the file system manager. The results of the change are sent to all nodes and the status of the command processing is returned to the node, and eventually, to the process issuing the command. For example, a command to add a disk to a file system originates on a user process and:

1. Is sent to the daemon and validated.
2. If acceptable, it is forwarded to the file system manager, which updates the file system descriptors.
3. All nodes that have this file system are notified of the need to refresh their cached copies of the file system descriptor.
4. The return code is forwarded to the originating daemon and then to the originating user process.

Note: This chain of communication might allow faults that are related to the processing of a command to occur on nodes other than the node on which the command was issued.

Application and user interaction with GPFS

There are four ways to interact with a GPFS file system.

You can interact with a GPFS file system by using any of the following interfaces:

- The IBM Spectrum Scale management GUI as described in [“IBM Spectrum Scale GUI”](#) on page 145.
- Operating system commands, which run at GPFS daemon initialization time or at file system mount time (as described in [“Operating system commands”](#) on page 18).
- Operating system calls, such as **open()**, from an application requiring access to a file that is controlled by GPFS (as described in [“Operating system calls”](#) on page 19).
- GPFS commands as described in *IBM Spectrum Scale: Command and Programming Reference*.
- GPFS programming interfaces as described in *IBM Spectrum Scale: Command and Programming Reference*.
- IBM Spectrum Scale management API as described in [“IBM Spectrum Scale management API”](#) on page 149.

Operating system commands

Operating system commands operate on GPFS data during the following scenarios.

- The initialization of the GPFS daemon
- The mounting of a file system

Initialization of the GPFS daemon

GPFS daemon initialization can be done automatically as part of the node startup sequence, or manually by using the `mmstartup` command.

The daemon startup process loads the necessary kernel extensions, if they are not previously loaded by another process. The daemon then waits for the cluster manager to declare that a quorum exists. When quorum is achieved, the cluster manager changes the state of the group from *initializing* to *active*. You can see the transition to active state when the `mmfsd ready` message appears in the GPFS log file (`/var/adm/ras/mmfs.log.latest`) or by running the `mmgetstate` command. When this state changes from *initializing* to *active*, the daemon is ready to accept mount requests.

The mounting of a file system

GPFS file systems are mounted by using the GPFS `mmmount` command.

On AIX or Linux, you can also use the operating system's `mount` command. GPFS mount processing builds the structures required to provide a path to the data and is performed on both the node requesting the mount and the file system manager node. If there is no file system manager, a call is made to the cluster manager, which appoints one. The file system manager ensures that the file system is ready to be mounted. The file system manager ensures that each of the following is true:

- There are no conflicting utilities being run by the **mmfsck** command, which checks and repairs a file system.
- There are no conflicting utilities being run by the **mmcheckquota** command, which checks file system user, group and fileset quotas.
- All of the disks are available.
- Any necessary file system log processing is completed to ensure that metadata on the file system is consistent.

On the local node, the control structures required for a mounted file system are initialized and the token management function domains are created. In addition, paths to each of the disks that make up the file system are opened. Part of mount processing involves unfencing the disks, which might be necessary if this node was previously failed. This is done automatically without user intervention. If insufficient disks are up, the mount fails. That is, in a replicated system if two disks are down in different failure groups, the mount fails. In a non-replicated system, one disk down causes the mount to fail.

Operating system calls

The most common interface to files residing in a GPFS file system is through normal file system calls to the operating system.

When a file is accessed, the operating system submits the request to the GPFS kernel extension, which attempts to satisfy the application request using data already in memory. If this can be accomplished, control is returned to the application through the operating system interface. If the data is not available in memory, the request is transferred for execution by a daemon thread. The daemon threads wait for work in a system call in the kernel, and are scheduled as necessary. Services available at the daemon level include the acquisition of tokens and disk I/O.

Operating system calls operate on GPFS data during:

- Opening of a file
- Reading of data
- Writing of data

Opening a GPFS file

The open of a file residing in a GPFS file system involves the application making a call to the operating system specifying the name of the file.

Processing of a file open involves two stages:

1. Identifying the file specified by the application.
2. Building the required data structures based on the inode.

The kernel extension code processes the directory search. If the required information is not in memory, the daemon is called to acquire the necessary tokens for the directory or part of the directory needed to resolve the lookup, then reads the directory from disk into memory.

The lookup process occurs one directory at a time in response to calls from the operating system. In the final stage of open, the inode for the file is read from disk and connected to the operating system vnode structure. This requires acquiring locks on the inode and a lock that indicates the presence to the metanode. The metanode is discovered or created any time a file is opened.

- If no other node has this file open, this node becomes the metanode.
- If another node has a previous open, then that node is the metanode and this node interfaces directly with the metanode for metadata operations.

If the open involves the creation of a new file, the appropriate locks are obtained on the parent directory and the inode allocation file block. The directory entry is created, an inode is selected and initialized and then open processing is completed.

Reading file data

The GPFS read function is invoked in response to a read system call.

File read processing falls into three levels of complexity based on system activity and status:

1. Buffers are available in memory.
2. Tokens are available in memory but data must be read.
3. Data and tokens must be acquired.

At the completion of a read, a determination of the need for prefetching is made. GPFS computes a desired read-ahead for each open file based on the performance of the disks, the data access pattern and the rate at which the application is reading data. If additional prefetching is needed, a message is sent to the daemon that processes it asynchronously with the completion of the current read.

Buffer and locks available in memory

The simplest read operation occurs when the data is already available in memory because either it is prefetched or because it has been read recently by another read call.

In either case, the buffer is locally locked and the data is copied to the application data area. The lock is released when the copy is complete. Note that no token communication is required because possession of the buffer implies that the node at least has a read token that includes the buffer. After the copying, prefetching is initiated if appropriate.

Tokens available locally but data must be read

The second, more complex, type of read operation is necessary when the data is not in memory.

This occurs under three conditions:

- The token is acquired on a previous read that found no contention.
- The buffer is stolen for other uses.
- On some random read operations.

In the first of a series of reads, the token is not available locally, but in the second read it might be available.

In such situations, the buffer is not found and must be read from disk. No token activity has occurred because the node has a sufficiently strong token to lock the required region of the file locally. A message is sent to the daemon, which is handled on one of the waiting daemon threads. The daemon allocates a buffer, locks the file range that is required so the token cannot be stolen for the duration of the I/O, and initiates the I/O to the device holding the data. The originating thread waits for this to complete and is posted by the daemon upon completion.

Data and tokens must be acquired

The third, and most complex read operation requires that tokens and data be acquired on the application node.

The kernel code determines that the data is not available locally and sends a message to the daemon waiting after posting the message. The daemon thread determines that it does not have the required tokens to perform the operation. In that case, a token acquire request is sent to the token management server. The requested token specifies a required length of that range of the file, which is needed for this buffer. If the file is being accessed sequentially, a desired range of data, starting at this point of this read and extending to the end of the file, is specified. In the event that no conflicts exist, the desired range is granted, eliminating the need for additional token calls on subsequent reads. After the minimum token needed is acquired, the flow proceeds with the token management process described in [“The file system manager”](#) on page 9.

Writing file data

write processing is initiated by a system call to the operating system, which calls GPFS when the write involves data in a GPFS file system.

GPFS moves data from a user buffer into a file system buffer synchronously with the application write call, but defers the actual write to disk. This asynchronous I/O technique allows better scheduling of the disk I/O operations and improved performance. The file system buffers come from the memory allocated based on the `pagepool` parameter in the `mmchconfig` command. Increasing the size of the pagepool might allow more writes to be deferred, which can improve performance in certain workloads.

A block of data is scheduled to be written to a disk when:

- The application has specified a synchronous write.
- The system needs the memory used to buffer the written data.
- The file token is revoked.
- The last byte of a block of a file is being written sequentially.
- A system sync command is run.

Until one of these occurs, the data remains in GPFS memory.

write processing falls into three levels of complexity based on system activity and status:

1. Buffer available in memory.
2. Tokens available locally but data must be read.
3. Data and tokens must be acquired.

Metadata changes are flushed under a subset of the same conditions. They can be written either directly, if this node is the metanode, or through the metanode, which merges changes from multiple nodes. This last case occurs most frequently if processes on multiple nodes are creating new data blocks in the same region of the file.

Buffer available in memory

The simplest path involves a case where a buffer already exists for this block of the file, but might not have a strong token.

This occurs if a previous write call accessed the block and it is still resident in memory. The write token already exists from the prior call. In this case, the data is copied from the application buffer to the GPFS buffer. If this is a sequential write and the last byte has been written, an asynchronous message is sent to the daemon to schedule the buffer for writing to disk. This operation occurs on the daemon thread overlapped with the execution of the application.

Token available locally but data must be read

There are two situations in which the token might exist but the buffer does not exist.

1. The buffer is recently stolen to satisfy other needs for buffer space.
2. A previous write obtained a desired range token for more than it needed.

In either case, the kernel extension determines that the buffer is not available, suspends the application thread, and sends a message to a daemon service thread requesting the buffer. If the write call is for a full file system block, an empty buffer is allocated since the entire block is replaced. If the write call is for less than a full block and the rest of the block exists, the existing version of the block must be read and overlaid. If the write call creates a new block in the file, the daemon searches the allocation map for a block that is free and assigns it to the file. With both a buffer assigned and a block on the disk associated with the buffer, the write proceeds as it would in [“Buffer available in memory” on page 21](#).

Data and tokens must be acquired

The third, and most complex path through write occurs when neither the buffer nor the token exists at the local node.

Prior to the allocation of a buffer, a token is acquired for the area of the file that is needed. As was true for read, if sequential operations are occurring, a token covering a larger range than is needed is obtained if no conflicts exist. If necessary, the token management function revokes the needed token from another node holding the token. Having acquired and locked the necessary token, the write continues as in [“Token available locally but data must be read” on page 21](#).

The stat() system call

The `stat()` system call returns data on the size and parameters associated with a file. The call is issued by the `ls -l` command and other similar functions.

The data required to satisfy the `stat()` system call is contained in the inode. GPFS processing of the `stat()` system call differs from other file systems in that it supports handling of `stat()` calls on all nodes without funneling the calls through a server.

This requires that GPFS obtain tokens that protect the accuracy of the metadata. To maximize parallelism, GPFS locks inodes individually and fetches individual inodes. In cases where a pattern can be detected, such as an attempt to `stat()` all of the files in a larger directory, inodes are fetched in parallel in anticipation of their use.

Inodes are cached within GPFS in two forms:

- Full inode
- Limited stat cache form

The full inode is required to perform data I/O against the file.

The stat cache form is smaller than the full inode, but is sufficient to open the file and satisfy a `stat()` call. It is intended to aid functions such as `ls -l`, `du`, and certain backup programs that scan entire directories looking for modification times and file sizes.

These caches and the requirement for individual tokens on inodes are the reason why a second invocation of directory scanning applications might run faster than the first.

GPFS command processing

GPFS commands fall into two categories: those that are processed locally and those that are processed at the file system manager for the file system involved in the command.

The file system manager is used to process any command that alters the state of the file system. When commands are issued and the file system is not mounted, a file system manager is appointed for the task. The `mmchdisk` command and the `mmfsck` command represent two typical types of commands that are processed at the file system manager.

The mmchdisk command

The **mmchdisk** command is issued when a failure that caused the unavailability of one or more disks has been corrected. The need for the command can be determined by the output of the **mmfsdisk** command.

The **mmchdisk** command performs four major functions:

- It changes the availability of the disk to `recovering`, and to `up` when all processing is complete. All GPFS utilities honor an availability of `down` and do not use the disk. `recovering` means that recovery has not been completed but the user has authorized use of the disk.
- It restores any replicas of data and metadata to their correct value. This involves scanning all metadata in the system and copying the latest to the recovering disk. Note that this involves scanning large amounts of data and potentially rewriting all data on the disk. This can take a long time for a large file system with a great deal of metadata to be scanned.
- It stops or suspends usage of a disk. This merely involves updating a disk state and should run quickly.
- Change disk attributes' metadata.

Subsequent invocations of the **mmchdisk** command attempts to restore the replicated data on any disk left in with an availability of `recovering`.

For more information, see *mmchdisk command* in *IBM Spectrum Scale: Command and Programming Reference*.

The mmfsck command

The **mmfsck** command repairs file system structures.

The **mmfsck** command operates in two modes:

1. online
2. offline

For performance reasons, GPFS logging allows the condition where disk blocks are marked **used** but not actually part of a file after a node failure. The online version of **mmfsck** cleans up that condition. Running `mmfsck -o -n` scans the file system to determine whether correction might be useful. The online version of **mmfsck** runs on the file system manager and scans all inodes and indirect blocks looking for disk blocks that are allocated but not used. If authorized to repair the file system, it releases the blocks. If not authorized to repair the file system, it reports the condition to standard output on the invoking node.

The offline version of **mmfsck** is the last line of defense for a file system that cannot be used. It is most often needed in the case where GPFS recovery logs are not available because of disk media failures. The **mmfsck** command runs on the file system manager and reports status to the invoking node. It is mutually incompatible with any other use of the file system and checks for any running commands or any nodes with the file system mounted. It exits if any are found. It also exits if any disks are **down** and require the use of **mmchdisk** to change them to **up** or **recovering**. The **mmfsck** command performs a full

file system scan looking for metadata inconsistencies. This process can be lengthy on large file systems. It seeks permission from the user to repair any problems that are found, which might result in the removal of corrupted files or directories. The processing of this command is similar to those for other file systems.

For more information, see *mmfsck command* in *IBM Spectrum Scale: Command and Programming Reference*.

NSD disk discovery

When the GPFS daemon starts on a node, it discovers the disks defined as NSDs by reading a disk descriptor that is written on each disk owned by GPFS. This enables the NSDs to be found regardless of the current operating system device name assigned to the disk.

On UNIX, NSD discovery is done by the GPFS shell script `/usr/lpp/mmfs/bin/mmdevdiscover`, which generates a list of available disk devices that appear in the node's local `/dev` file system. On Linux, any devices designated to be an IBM Spectrum Scale NSD must also exist in the `/proc/partitions` directory. Each NSD is assigned a device type as shown in the following tables:

Table 3. Supported devices on AIX		
Device Name	GPFS Device Type	Description
vpath	vpath	IBM virtual path disk
hdisk	hdisk	AIX hard disk
hdiskpower	powerdisk	EMC power path disk
dlnmfdrv	dlnmfdrv	Hitachi dlnm

Table 4. Supported devices on Linux		
Device Name	GPFS Device Type	Description
dm-	dmm	Device-Mapper Multipath (DMM)
vpath	vpath	IBM virtual path disk
sd/hd	Generic	Device having no unique failover or multipathing characteristic (predominantly Linux devices).
emcpower	powerdisk	EMC power path disk
dsad*	dsad	Direct access storage devices (DASD) (for Linux on z Systems)
vd*	Generic	Device having no unique failover or multipathing characteristic (predominantly Linux devices).
sd*	Generic	SCSI disk.
hd*	Generic	Disk on IDE controller.
scini*	Generic	Device having no unique failover or multipathing characteristic (predominantly Linux devices).
pmem*	Generic	Device having no unique failover or multipathing characteristic (predominantly Linux devices).
nvm*	Generic	NVMe disk.

Table 5. Supported devices on Windows		
Device Name	GPFS Device Type	Description
A Number 0-n	gpt	GPFS partition on Windows disk

To override or enhance NSD discovery, you can create a script and name it `/var/mmfs/etc/nsddevices`. The user-created `nsddevices` script, if it exists, is executed before the default discovery process.

On Windows, NSDs have a GUID Partition Table (GPT) with a single GPFS partition. NSD discovery is done by scanning the system for a list of disks that contain a GPFS partition.

On all platforms, the list of disk devices is then used by the GPFS daemon to determine whether a device interface on the local node maps to an NSD name recorded in the configuration database. The process of mapping a device interface to an NSD involves GPFS opening each device in turn and reading any NSD volume identifier potentially recorded at sector two of the disk.

If GPFS discovers that an NSD volume identifier read from a disk device matches the volume identifier recorded with the NSD name in the GPFS configuration database, I/O for the local node proceeds over the local device interface.

If no device mapping appears on the local node for a particular NSD, I/O proceeds over the IP network to the first NSD server specified in the server list for that NSD. If the first NSD server in the server list is not available, I/O proceeds sequentially through the server list until it finds an available NSD server.

Consult the `/usr/lpp/mmfs/samples/nsddevices.sample` file for configuration guidelines on how to provide additional disk discovery capabilities unique to your configuration.

Failure recovery processing

GPFS failure recovery processing occurs automatically. Therefore, though not necessary but some familiarity with its internal functions is useful when failures are observed.

Only one state change, such as the loss or initialization of a node, can be processed at a time and subsequent changes are queued. Therefore, the entire failure processing must be completed before the failed node can rejoin the group. All failures are processed first, which means that GPFS handles all failures before it completes any recovery.

GPFS recovers from a node failure by using join or leave processing messages that are sent explicitly by the cluster manager node. The cluster manager node observes that a node fails when it no longer receives heartbeat messages from the node. The join or leave processing messages are broadcast to the entire group of nodes that run GPFS, and each node updates its status for the failing or joining node. Failure of the cluster manager node results in a new cluster manager that is elected by the cluster. Then, the newly elected cluster configuration manager node processes the failure message for the failed cluster manager.

GPFS is notified of a node failure or that the GPFS daemon has failed on a node. It then starts the recovery for each of the file systems that were mounted on the failed node. If necessary, new file system managers are selected for any file systems that no longer have one.

The file system manager for each file system ensures that the failed node can no longer access the disks that comprise the file system. If the file system manager is newly appointed as a result of this failure, it rebuilds a token state by querying the other nodes in the group. After this rebuilding process is completed, the actual recovery of the failed node log, begins. This recovery also rebuilds the metadata that was being modified at the time of the failure to a consistent state. In some cases, blocks are allocated that are not part of any file and are effectively lost until `mmfsck` is run, online, or offline. After log recovery is complete, the locks that are held by the failed nodes are released for this file system. When this activity is completed for all file systems, failure processing is done. The last step of this process allows a failed node to rejoin the cluster.

Cluster configuration data files

GPFS commands store configuration and file system information in one or more files collectively known as GPFS cluster configuration data files. These files are not intended to be modified manually.

The GPFS administration commands are designed to keep these files synchronized between each other and with the GPFS system files on each node in the cluster. The GPFS commands constantly update the GPFS cluster configuration data files and any user modification made to this information might be lost without warning. On AIX nodes this includes the GPFS file system stanzas in **/etc/filesystems** and on Linux nodes the lists in **/etc/fstab**.

The GPFS cluster configuration data is stored in the `/var/mmfs/gen/mmsdrfs` file. The first record in the `mmsdrfs` file contains a generation number. Whenever a GPFS command causes something to change in the cluster or any of the file systems, this change is reflected in the `mmsdrfs` file and the generation number is increased by one. The latest generation number is always recorded in the `mmsdrfs` file on CCR.

The master copy of configuration data files is stored redundantly on all quorum nodes. This method of storing configuration data is called the cluster configuration repository (CCR) and is the default for new clusters created on IBM Spectrum Scale. Existing clusters can be converted to the new repository type using the `--ccr-enable` option of the `mmchcluster` command.

Using CCR has the advantage that full read/write access to the configuration data remains available as long as a majority of quorum nodes are accessible. For example, in a cluster with five quorum nodes, commands that update the `mmsdrfs` file continues to work normally, even if any two of the five quorum nodes have failed. In a two-node cluster with tiebreaker disks, it is still possible to run commands that change the `mmsdrfs` file if one of the two nodes has failed, as long as the surviving node has access to the tiebreaker disks. In general, full configuration command functionality remains available as long as enough nodes and, if specified, tiebreaker disks are accessible for GPFS to reach quorum. The CCR also has the advantage that it allows changing the tiebreaker disk configuration, including switching between node-based quorum and node quorum with tiebreaker, without first shutting down GPFS on all of the nodes. For more information about CCR, see [“Cluster configuration repository”](#) on page 26.

Based on the information in the GPFS cluster configuration data, the GPFS commands generate and maintain a number of system files on each of the nodes in the GPFS cluster.

Linux

`/etc/fstab`

On Linux nodes, contains lists for all GPFS file systems that exist in the cluster.

AIX

`/etc/filesystems`

On AIX nodes, contains lists for all GPFS file systems that exist in the cluster.

All GPFS nodes

`/var/mmfs/gen/mmfsNodeData`

Contains GPFS cluster configuration data pertaining to the node.

`/var/mmfs/gen/mmsdrfs`

Contains a local copy of the `mmsdrfs` file from CCR.

`/var/mmfs/gen/mmfs.cfg`

Contains GPFS daemon startup parameters.

GPFS backup data

The GPFS `mmbbackup` command creates several files during command execution. Some of the files are temporary and deleted at the end of the backup operation. There are other files that remain in the root directory of the fileset or the file system and they must not be deleted.

The `mmbbackup` command creates other files that begin with `.mmbbackupShadow.*`. These files are associated with the `mmbbackup` command and are required for proper backups to be complete, so do not manually delete or change them.

For more information, see *mmbbackup command* in *IBM Spectrum Scale: Command and Programming Reference*.

Cluster configuration repository

The cluster configuration repository (CCR) of IBM Spectrum Scale is a fault tolerant configuration store that is used by nearly all IBM Spectrum Scale components, including GPFS, GUI, system health, and Cluster Export Services (CES) to name a few. It is not meant to be used directly by the customer. It offers an interface to store configuration files and flat key-value pairs.

The CCR state consists of files and directories under `/var/mmfs/ccr`. The consistency of CCR state replicas is maintained by using a majority consensus algorithm. That is, the CCR state from a majority of quorum nodes or tiebreaker disks must be available for the CCR to function properly. For example, a cluster with just one or two quorum nodes and no tiebreaker disks has zero fault tolerance. If CCR state is unavailable on one of the quorum nodes, the cluster reports a quorum loss. If the CCR state is unavailable on both quorum nodes, the cluster becomes inoperable. You can create a fault tolerance by assigning more quorum nodes or tiebreaker disks.

Tiebreaker disks can be configured while GPFS is up and running. Increasing the fault tolerance by leveraging up to eight quorum nodes and up to three tiebreaker disks helps to keep the CCR state. That means the CCR remains fully functional as long as a majority of quorum nodes or tiebreaker disks are available.

Common CCR functions

The CCR has the following functions:

- Provides a PUT and GET interface for storing configuration files and flat key-value pairs redundantly across all quorum nodes. CCR uses a Paxos based algorithm to keep the stored data consistent among the quorum nodes. That is, ensure that all quorum nodes agree on the most recent version of each configuration file or key-value pair.
- Updates the CCR configuration when the number of quorum nodes or tiebreaker disks are changing.
- Creates a CCR backup file and initializes the entire quorum nodes from a CCR backup by using the **`mmsdrrestore -F <CCR_BACKUP_FILE> -a`** command.
- Elects the cluster manager by running the Paxos protocol among the available quorum nodes.

If the cluster is configured with tiebreaker disks and the network between the current cluster manager and the remaining quorum nodes is not working, then one of the quorum nodes writes the so-called challenges to the tiebreaker disks. The current cluster manager must answer within a specific time limit to keep this role. Otherwise, the challenger node becomes the new cluster manager.

- Provides monitor and debug commands that can be used for analysis when CCR is not functional. For more information about how to use these commands, see *CCR issues* in *IBM Spectrum Scale: Problem Determination Guide*.

GPUDirect Storage support for IBM Spectrum Scale

IBM Spectrum Scale's support for NVIDIA's GPUDirect Storage (GDS) enables a direct path between GPU memory and storage. This solution addresses the need for higher throughput and lower latencies. File system storage is directly connected to the GPU buffers to reduce latency and load on CPU. For IBM

Spectrum Scale, this means that data can be read directly from an NSD server's pagepool and it is sent to the GPU buffer of the IBM Spectrum Scale clients by using RDMA. IBM Spectrum Scale with GDS requires an InfiniBand or RoCE fabric. In IBM Spectrum Scale, the **mmdiag** command is enhanced to print diagnostic information for GPUDirect Storage.

IBM Spectrum Scale supports Nvidia's GDS over RDMA over Converged Ethernet (RoCE). It allows to achieve low latencies and high throughput when reading data from an NSD server pagepool into a GPU buffer. It requires a high-speed Ethernet fabric with GDS capable hardware and the CUDA environment must be installed on GDS clients.

GDS is useful where significant I/O is involved and the CPU stands as a bottleneck to overall system performance. This happens when the CPU cycles are heavily used for managing data transfers into and out of CPU memory and CPU DRAM bandwidth is used. The addition of GDS enhances the ability of IBM Spectrum Scale and all-flash solutions like ESS 3200 to avoid many of those bottlenecks.

You need to install CUDA, which is provided by NVIDIA, on the IBM Spectrum Scale client. GDS enables the CUDA developer to bring data directly from IBM Spectrum Scale storage to the GPU memory by using RDMA. GDS eliminates the buffer copies in system (CPU) memory, bypasses the CPU memory, and places data directly into the GPU application memory. GDS delivers benefits such as increased data transfer rate, lower latency, and reduced CPU utilization.

I/O requests triggered by calling the function `cuFileRead()` from the CUDA application are run as RDMA requests from the NSD servers into the GPU buffer. Currently, only read operations are supported as GDS/IO (`cuFileRead()`). Writes operations (`cuFileWrite()`) are handled transparently in compatibility mode.

Compatibility mode

For certain types of I/O, GDS cannot use the direct RDMA from the pagepool into the GPU buffer. In those cases, the buffered I/O path is taken, which gets the data correctly into the GPU but it does not produce any performance improvements. This is called *compatibility mode* for GDS. The types of I/O that switches GDS into compatibility mode are as follows:

- Files with size less than 4096 bytes.
- Sparse files or files with preallocated storage. For example, `fallocate()` and `gpfs_prealloc()`.
- Encrypted files.
- Memory-mapped files.
- Compressed files or files that are marked for deferred compression. For more information on compression, see *File compression* in *IBM Spectrum Scale: Administration Guide*.
- Files in snapshots or clones.
- Direct I/O is disabled by using the `mmchconfig disableDIO = true` option. The default value of the **disableDIO** parameter is `false`.

IBM Spectrum Scale supports recovery for failing GDS RDMA operations by returning the failed read request to CUDA and CUDA retries the failed read request in the compatibility mode.

Other limitations

The following limitations are also applicable for the GDS support:

- Write operations (`cuFileWrite`) are supported in compatibility mode. For the writes in compatibility mode, the data is first copied from the GPU buffer into the host memory (`cudaMemcpy`) and then Direct I/O is used if possible.
- IBM Spectrum Scale does not support GDS in the following scenarios:
 - NVIDIA GDS in asynchronous "poll" mode. The NVIDIA GDS lib implicitly converts a poll mode request on a file in an IBM Spectrum Scale mount to a synchronous GDS I/O request.
 - Reading a file with GDS read concurrently with a buffered read does not deliver full GDS performance for the GDS thread. This limitation holds whether the concurrent threads are part of the same or different user application. In this context, *buffered read* is considered as a nonGDS and indirect I/O.

- Files that use data tiering, including Transparent Cloud Tiering.

Related concepts

[“Planning for GPUDirect Storage” on page 250](#)

IBM Spectrum Scale support for GPUDirect Storage (GDS) enables a direct path between GPU memory and storage. You need to ensure that certain conditions are met before you start installing the feature.

Related tasks

[“Installing GPUDirect Storage for IBM Spectrum Scale” on page 481](#)

IBM Spectrum Scale support for GPUDirect Storage (GDS) enables a direct path between GPU memory and storage.

[“Upgrading GPUDirect Storage” on page 517](#)

You need to upgrade your IBM Spectrum Scale cluster to 5.1.2 or later to start using the GPUDirect Storage (GDS).

Related information

Protocols support overview: Integration of protocol access methods with GPFS

IBM Spectrum Scale provides extra protocol access methods. Providing these additional file and object access methods and integrating them with GPFS offers several benefits. It enables users to consolidate various sources of data efficiently in one global namespace. It provides a unified data management solution and enables not only efficient space utilization but also avoids making unnecessary data moves just because access methods might be different.

Protocol access methods that are integrated with GPFS are NFS, SMB, HDFS, and Object. While each of these server functions (NFS, SMB, HDFS, and Object) uses open source technologies, this integration adds value by providing scaling and by providing high availability by using the clustering technology in GPFS.

For information about HDFS protocol support, see *CES HDFS in IBM Spectrum Scale Big Data and Analytics Support* documentation.

- The integration of file and object serving with GPFS allows the ability to create NFS exports, SMB shares, and Object containers that have data in GPFS file systems for access by client systems that do not run GPFS.
- Some nodes (at least two recommended) in the GPFS cluster must be designated as protocol nodes (also called CES nodes) from which (non-GPFS) clients can access data that resides in and is managed by GPFS by using the appropriate protocol artifacts (exports, shares, or containers).
- The protocol nodes need to have GPFS server license designations.
- The protocol nodes must be configured with "external" network addresses that are used to access the protocol artifacts from clients. The (external) network addresses are different from the GPFS cluster address that is used to add the protocol nodes to the GPFS cluster.
- The CES nodes allow SMB, NFS, HDFS, and Object clients to access the file system data using the protocol exports on the configured public IP addresses. The CES framework allows network addresses that are associated with protocol nodes to fail over to other protocol nodes when a protocol node fails.

All protocol nodes in a cluster must be running on the same operating system and they must be of the same CPU architecture. The other nodes in the GPFS cluster might be on other platforms and operating systems.

For information about supported operating systems for protocol nodes and their required minimum kernel levels, see [IBM Spectrum Scale FAQ in IBM Documentation](#).

Like GPFS, the protocol serving functionality is also delivered only as software.

- The CES framework provides access to data managed by GPFS through more access methods.

- While the protocol function provides several aspects of NAS file serving, the delivery is not a NAS appliance.
- Role-based access control of the command line interface is not offered.
- Further, the types of workloads that are suited for this delivery continue to be workloads that require the scaling or consolidation aspects that are associated with traditional GPFS.

Note: Some NAS workloads might not be suited for delivery in the current release. For instance, extensive use of snapshots, or support for many SMB users. For SMB limitations, see [“SMB limitations” on page 306](#).

For more information, see [IBM Spectrum Scale FAQ in IBM Documentation](#).

Note: Even though some of the components that are provided are open source, the specific packages that are provided must be used. If there are existing versions of these open source packages on your system, they must be removed before you install this software.

The protocols feature provides following additional commands for administering the protocols, along with the enhancements to the existing commands:

- The commands for managing these functions includes `mmces`, `mmuserauth`, `mmnfs`, `mmsmb`, `mmobj`, and `mmperfmon`.
- In addition, `mmdumpperfdata` and `mmprotocoltrace` are provided to help with data collection and tracing.
- Existing GPFS commands are expanded with some options for protocols include `mmclscluster`, `mmchnode`, and `mmchconfig`.
- The `gpfs.snap` command is enhanced to include data by gathering about the protocols to help with problem determination.

For information on the use of CES including administering and managing the protocols, see the *Implementing Cluster Export Services* chapter of *IBM Spectrum Scale: Administration Guide*.

In addition to the installation toolkit, IBM Spectrum Scale also includes a performance monitoring toolkit. Sensors to collect performance information are installed on all protocol nodes, having one of these nodes as a designated collector node. The `mmperfmon query` command can be used to view the performance counters that have been collected.

The `mmhealth` command can be used to monitor the health of the node and the services hosted on that node.

Cluster Export Services overview

Cluster Export Services (CES) provides highly available file and object services to a GPFS cluster by using Network File System (NFS), Hadoop Distributed File System (HDFS), Object, or Server Message Block (SMB) protocols.

IBM Spectrum Scale includes Cluster Export Services (CES) infrastructure to support the integration of the NFS, HDFS, SMB, and Object servers.

- The NFS server supports NFS v3 and the mandatory features in NFS v4.0 and NFS v4.1.
- The SMB server supports SMB 2, SMB 2.1, and the mandatory features of SMB 3.0 and 3.11.
- The Object server supports the Train release of OpenStack Swift along with Keystone v3.
- CES supports HDFS protocols. HDFS Transparency is supported from IBM Spectrum Scale version 3.1.1. The IBM Spectrum Scale Big Data Analytics Integration Toolkit for HDFS Transparency or Toolkit for HDFS must be installed. The Toolkit for HDFS is supported from version 1.0.0.0. See [Support Matrix](#) under *CES HDFS in Big data and analytics support documentation*.

The CES infrastructure is responsible for:

- Managing the setup for high-availability clustering that is used by the protocols.

- Monitoring the health of these protocols on the protocol nodes and raising events or alerts during failures.
- Managing the addresses that are used for accessing these protocols by including failover and failback of these addresses because of protocol node failures.

High availability

With GPFS, you can configure a subset of nodes in the cluster to provide a highly available solution for exporting GPFS file systems by using the Network File System (NFS), Server Message Block (SMB), Hadoop Distributed File System (HDFS), and Object protocols. The participating nodes are designated as Cluster Export Services (CES) nodes or protocol nodes. The set of CES nodes is frequently referred to as the *CES cluster*.

A set of IP addresses, the *CES address pool*, is defined and distributed among the CES nodes. As nodes enter and leave the GPFS cluster, the addresses in the pool can be redistributed among the CES nodes to provide high availability. GPFS nodes that are not CES nodes that are leaving or entering the GPFS cluster do not redistribute the IP addresses. It is possible to use one IP address for all CES services. However, clients that use the SMB, NFS, and Object protocols must not share the IP address for these protocols with the IP addresses that are used by the HDFS service to avoid impacting the clients of other protocols during an HDFS failover. However, applications that use the CES IP addresses are not impacted. SMB failover is not transparent.

Monitoring

CES monitors the state of the protocol services. Monitoring ensures that the CES addresses are assigned to the appropriate node and that the processes that implement the services in the CES cluster are running correctly. Upon failure detection, CES marks the node as temporarily unable to participate in the CES cluster and reassigns the IP addresses to another node.

Protocol support

CES supports the following export protocols: NFS, SMB, HDFS, and Object. Each protocol can be enabled or disabled in the cluster. If a protocol is enabled in the CES cluster, all CES nodes serve that protocol.

The following are examples of enabling and disabling protocol services by using the **mmces** command:

mmces service enable nfs

Enables the NFS protocol in the CES cluster.

mmces service disable obj

Disables the Object protocol in the CES cluster.

Commands

To set or configure CES options, the following commands are used:

mmces

Manages the CES address pool and other CES cluster configuration options.

mmhdfs

Manages the HDFS configuration operations.

mmnfs

Manages NFS exports and sets the NFS configuration attributes.

mmobj

Manages the Object configuration operations.

mm smb

Manages SMB exports and sets the SMB configuration attributes.

mmuserauth

Configures the authentication methods that are used by the protocols.

For more information, see *mmces* command, *mmhdfs*, *mmnfs* command, *mmobj* command, *mmsmb* command, and *mmuserauth* command in *IBM Spectrum Scale: Command and Programming Reference*.

Restrictions

For an up-to-date list of supported operating systems, specific distributions, and other dependencies for CES, refer to the [IBM Spectrum Scale FAQ](#) in IBM Documentation.

A CES cluster can contain a maximum of 32 CES nodes. All the nodes in a CES cluster must be running on the same operating system and they must be of the same CPU architecture. If the SMB protocol is enabled, then the CES cluster is limited to a total of 16 CES nodes.

Each CES node must have network adapters capable of supporting all IP addresses in the CES address pool. The primary address of these adapters must not be used as a CES address.

NFS support overview

The NFS support for IBM Spectrum Scale enables clients to access the GPFS file system by using NFS clients with their inherent NFS semantics.

The following features are provided:

Clustered NFS Support

NFS clients can connect to any of the protocol nodes and get access to the exports defined. A clustered registry makes sure that all nodes see the same configuration data. It does not matter to the client to which of the CES nodes the connections are established. Moreover, the state of opened files is also shared among the CES nodes so that data integrity is maintained. On failures, clients can reconnect to another cluster node as the IP addresses of failing nodes are automatically transferred to another healthy cluster node. The supported protocol levels are NFS version 3 (NFSv3) and NFS version 4 (NFSv4.0, NFSv4.1).

Export management commands

With the **mmnfs export** command, IBM Spectrum Scale provides a comprehensive entry point to manage all NFS-related export tasks such as creating, changing, and deleting NFS exports. The **mmnfs export** command follows the notions of supported options, that is, a limited set of NFS-related options that proved useful. For more information, see the *mmnfs* command in the *IBM Spectrum Scale: Command and Programming Reference*.

Export configuration commands

With the **mmnfs config** command, IBM Spectrum Scale provides a tool for administering the global NFS configuration. You can use this command to set and list default settings such as the port number for the NFS service, the default access mode for exported file systems, and the NFS server log level. This command follows the notions of supported options, that is, a limited set of NFS-related options that proved useful. For more information, see the *mmnfs* command in the *IBM Spectrum Scale: Command and Programming Reference*.

NFS monitoring

The monitoring framework detects issues with the NFS services and triggers failover if an unrecoverable error occurs. Moreover, the **mmces** command provides a quick access to current and past system states and these states are useful to diagnose issues with the NFS services on the CES nodes. You can check the node performance by using the **mmhealth node show** command. Issues that are detected and causing failover are, for example, GPFS daemon failures, node failures, or NFS service failures. For more information, see the *mmces* command and the *mmhealth* command in *IBM Spectrum Scale: Command and Programming Reference*.

NFS performance metrics

The NFS services provide performance metrics that are collected by the performance monitor framework. The **mmperfmon query** tool provides access to the most important NFS metrics through predefined queries. For more information, see the **mmperfmon** command in the *IBM Spectrum Scale: Command and Programming Reference*.

Cross-protocol integration with SMB

IBM Spectrum Scale enables concurrent access to the same file data by using NFS, SMB, and native POSIX access (limitations apply).

Authentication and ID mapping

You can configure NFS services to authenticate against the most popular authentication services such as 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 or by using the external ID-mapping service like RFC 2307. If none of the offered authentication and mapping schemes match the environmental requirements, the option to establish a user-defined configuration is available. The **mmuserauth service create** command can be used to set up all authentication-related settings. For more information, see the **mmuserauth** command in the *IBM Spectrum Scale: Command and Programming Reference*.

SMB support overview

The SMB support for IBM Spectrum Scale allows clients to access the GPFS file system by using SMB clients with their inherent SMB semantics.

The following features are provided:

- **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, that is, for the client it does not matter to which of the CES nodes the connection is established. Moreover, the state of opened files (share modes, open modes, access masks, and locks) is also shared among the CES nodes so that data integrity is maintained. On failures, clients can reconnect to another cluster node as the IP addresses of failing nodes are transferred to another healthy cluster node. The supported protocol levels are SMB2 and the base functionality of SMB3 (dialect negotiation, secure negotiation, encryption of data on the wire).

- **Export Management command**

With the **mm smb** command, IBM Spectrum Scale provides a comprehensive entry point to manage all SMB-related configuration tasks like creating, changing, and deleting SMB shares and administering the global configuration. The **mm smb** command follows the notions of supported options, that is, a limited set of SMB-related options that are proven useful. Moreover, the Microsoft Management Console can be used to administer SMB shares.

- **SMB monitoring**

The monitoring framework detects issues with the SMB services and triggers failover if there is an unrecoverable error. Moreover, the **mmhealth** and **mmces** commands provide quick access to current and past system states and aid to diagnose issues with the SMB services on the CES nodes. Issues that are detected and causing failover are, for example, GPFS daemon failures, node failures, or SMB service failures.

- **Integrated installation**

The SMB services can be installed by the integrated installer (toolkit) together with the CES framework, and the NFS and Object protocols.

- **SMB performance metrics**

The SMB services provide two sets of performance metrics that are collected by the performance monitor framework. Thus, not just the current metrics can be retrieved but also historic data is available (at some lower granularity). The two sets of metrics are global SMB metrics (like number of connects and disconnects) and the metrics for each SMB request (number, time, throughput). The **mmperfmon query** tool provides access to all SMB metrics by using predefined queries. Moreover, metrics for the clustered file metadata base CTDB are collected and exposed by using the **mmperfmon query** command.

- **Cross-protocol integration with NFS and SMB**

IBM Spectrum Scale allows concurrent access to the same file data by using unified file (such as, SMB and NFS), and object interfaces. For more information, see [“Unified file and object access overview” on page 34](#).

- **Authentication and ID mapping**

The SMB services can be configured to authenticate against the most popular authentication services MS Active Directory and LDAP. Mapping MS security identifiers (SIDs) to the POSIX user and group IDs on the file server can either be done automatically by using the automatic or by using an external ID mapping service like RFC 2307. If none of the offered authentication and mapping schemes matches the environmental requirements the option to establish a user-defined configuration is available. The **mmuserauth service create** command can be used to set up all authentication-related settings.

Object storage support overview

IBM Spectrum Scale object storage combines the benefits of IBM Spectrum Scale with the most widely used open source object store, OpenStack Swift.

Data centers are currently struggling to store and manage vast amounts of data efficiently and cost-effectively. The increasing number of application domains create silos of storage within data centers. These application domains include analytics, online transaction processing (OLTP), and high-performance computing (HPC). With each new application, a new storage system can be required, forcing system administrators to become experts in numerous storage management tools.

In addition, the set of applications that includes mobile and web-based applications, archiving, backup, and cloud storage also has another type of storage system for the system administrator to manage - object storage. Objects cannot be updated after they are created (although they can be replaced, versioned, or removed), and in many cases the objects are accessible in an eventually consistent manner. These types of semantics are suited for:

- Images
- Videos
- Text documents
- Virtual machine (VM) images
- Other similar files

IBM Spectrum Scale object storage combines the benefits of IBM Spectrum Scale with the most widely used open source object store today, OpenStack Swift. In IBM Spectrum Scale object storage, data is managed as objects and it can be accessed over the network by using RESTful HTTP-based APIs. This object storage system uses a distributed architecture with no central point of control, providing greater scalability and redundancy. IBM Spectrum Scale object storage organizes data in the following hierarchy:

1. Account

An account is used to group or isolate resources. Each object user is part of an account. Object users are assigned to a project (account) in keystone and can access only the objects that are within or part of the account. Each user is assigned to a project with a role that defines with the user rights and privileges to perform a specific set of operations on the resources of the account to which it belongs.

Users can be part of multiple accounts but it is mandatory that a user must be associated with at least one account. You must create at least one account before you add users. Account contains a list of containers in the object storage. You can also define quota at the account level.

Note: There is a one-to-one mapping between accounts in the object protocol and the keystone projects.

2. Container

Containers contain objects. Each container maintains a list of objects and provides a namespace for the objects. You can create any number of containers within an account.

3. Object

Objects store data. You can create, modify, and delete objects. Accounts have containers, and containers store objects. Each object is accessed through a unique URL. The URLs in the API contain the storage account, container name, and object name. You can define quota both at the account and container levels.

IBM Spectrum Scale object storage also offers the following features.

Storage policies for Object Storage

Storage policies enable segmenting of the Object Storage within a single cluster for various use cases.

Currently, OpenStack Swift supports storage policies in which you can define the replication settings and location of objects in a cluster. For more information, see *Open Stack Documentation for Storage Policies* in *IBM Spectrum Scale: Administration Guide*. IBM Spectrum Scale enhances storage policies to add the following functions for Object Storage:

- The file-access object capability (unified file and object access)
- Compression
- Encryption

You can use the **mmobj policy create** command to create a storage policy with specified functions from the available options. After a storage policy is created, you can specify that storage policy while you create new containers to associate that storage policy with those containers.

A fileset is associated with the new storage policy. You can (optionally) provide the name of the fileset as an argument of the **mmobj policy create** command. You can also map a storage policy and fileset. For more information, see *Mapping of storage policies to filesets* in *IBM Spectrum Scale: Administration Guide*.

You can also create, list, and change storage policies. For more information, see *Administering storage policies for Object Storage* and *mmobj command* in *IBM Spectrum Scale: Administration Guide* and *IBM Spectrum Scale: Command and Programming Reference*.

Unified file and object access overview

Unified file and object access allows use cases in which you can access data by using object or file interfaces.

Some of the key unified file and object access use cases are:

- You can use object stores to store large amounts of data. Object stores are highly scalable and are an economical storage solution. To analyze large amounts of data, advanced analytics systems are used. However, porting the data from an object store to a distributed file system that the analytics system requires is complex and time intensive. These scenarios reflect a need to access the object data by using the file interface so that analytics systems can use that data.
- Accessing object by using file interfaces and accessing file by using object interfaces helps legacy applications that are designed for file to start integrating into the object world after data migration.
- It allows storage cloud data that is in form of objects to be accessed by using files from applications that are designed to process files.
- It allows files that are exported by using Network File System (NFS) or Server Message Block (SMB) to be accessible as objects by using Hypertext Transfer Protocol (HTTP) to the end clients. It also allows files available on POSIX to be accessible as objects by using HTTP to the end clients. This enables easy availability of file data on mobile devices such as smartphones or tablets that are more suited to REST-based interfaces.

- Multi-protocol access for file and object in the same namespace allows supporting and hosting **data oceans** of different types with multiple access options.

For information about data oceans, see [“Protocols support overview: Integration of protocol access methods with GPFS”](#) on page 28.

- There is a rich set of placement policies for files (by using **mmapplypolicy**) available with IBM Spectrum Scale. With unified file and object access, those placement policies can be used for object data.

Unified file and object access allows users to access the same data as an object and as a file. Data can be stored and retrieved through IBM Spectrum Scale for Object Storage or as files from POSIX, NFS, and SMB interfaces. Unified file and object access provides the following capabilities to users:

- Ingest data through the object interface and access this data from the file interface.
- Ingest data through the file interface and access this data from the object interface.
- Ingest and access same data through object and file interfaces concurrently.
- Manage authentication and authorization in unified file and object access.

For more information, see *Unified file and object access in IBM Spectrum Scale* in *IBM Spectrum Scale: Administration Guide*.

One of the key advantages of unified file and object access is the placement and naming of objects when stored on the file system. For more information, see *File path in unified file and object access* in *IBM Spectrum Scale: Administration Guide*.

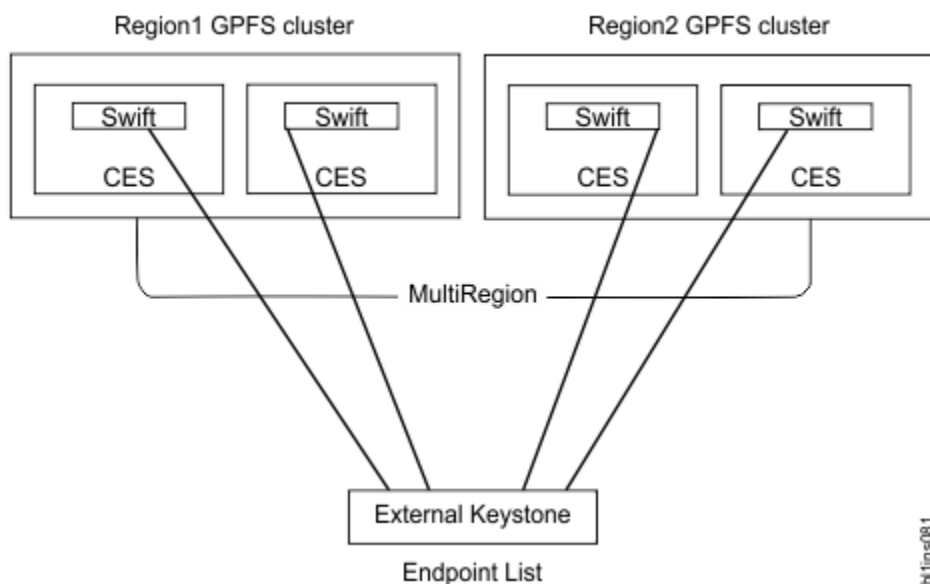
Overview of multi-region object deployment

Object multi-region deployment can be used to distribute the object load over several sites to reduce latency for local requests. It can also be used to provide an active-active disaster recovery configuration.

The main purpose of the object protocol is to enable the upload and download of object data. When clients have a fast connection to the cluster, the network delay is minimal. However, when client access to object data is over a WAN or a high-latency network, the network can introduce an unacceptable delay and affect quality of service metrics. To improve that response time, you can create a replica of the data in a cluster closer to the clients by using the active-active multi-region replication support in OpenStack Swift. Multi-region can also be used to distribute the object load over several clusters to reduce contention in the file system.

With multi-region support in Swift, the replication servers asynchronously copy data between remote clusters to keep them in sync. When the client uploads or downloads objects, the proxy server that the client connects to initially uses the devices in the same region as the proxy.

If the proxy cannot find the data that it needs in its own region, it then uses the other regions. In this way, the clients generally get fast access for the data in the close cluster. The clients also are affected by only the high-latency penalty when they access data that has not been replicated.



Multi-region object deployment involves creating up to 3 independent CES clusters. Each CES cluster is a single region. The Swift environment and ring files are configured to map each cluster to an associated region index. The configuration is then synced manually between the clusters to enable the active-active replication.

Enabling multi-region support converts the underlying primary Swift account, container, and object rings to include all defined regions. By default, all data is stored in all regions. To store data in just a subset of the regions, storage policies can be used to create object rings that store the objects in a subset of the regions. For more information on creating custom storage policies, see *mmobj* command in *IBM Spectrum Scale: Command and Programming Reference*.

Only the first cluster can switch to multi-cluster after installation. Subsequent clusters need to be installed as multi-cluster environments due to the need for region numbers and storage policy indexes to be globally consistent across clusters.

For information on planning a multi-region object deployment, see [“Planning for multi-region object deployment”](#) on page 326.

For information on enabling multi-region object deployment, see [“Enabling multi-region object deployment initially”](#) on page 448.

For information on adding a region to a multi-region object deployment environment, see *Adding a region in a multi-region object deployment* in *IBM Spectrum Scale: Administration Guide*.

S3 API

IBM Spectrum Scale Object Storage supports the S3 API, in addition to Swift API, for accessing object data.

IBM Spectrum Scale uses the s3api middleware for OpenStack Swift, allowing access to IBM Spectrum Scale by using the Amazon Simple Storage Service (S3) API. IBM Spectrum Scale for Object Storage includes S3 API as an optional feature.

S3 API can be enabled during protocol deployment, initial object configuration, or later on.

- For information on enabling S3 API during protocol deployment by using the `-s3` option of the **spectrumscale config object** command, see [“Deploying protocols”](#) on page 416.
- For information on enabling S3 API during initial object configuration by using the `--enable-s3` option of the **mmobj swift base** command, see *Configuring and enabling the Object protocol service* in *IBM Spectrum Scale: Administration Guide*.
- For information on enabling S3 API if it is not enabled as part of the object base configuration, see *Changing the object base configuration to enable S3 API* in *IBM Spectrum Scale: Administration Guide*.

Accessing the Object Storage through swift requests is not affected by enabling the S3 API. When the S3 API is enabled, the object service also recognizes S3 API requests sent to the TCP port used by the object service (8080).

For more information on S3 API, see the S3 API documentation at [Amazon S3 REST API Introduction](#).

For limitations of the S3 API support with IBM Spectrum Scale, see *Managing OpenStack access control lists using S3 API* in *IBM Spectrum Scale: Administration Guide*.

Object capabilities

Object capabilities describe the object protocol features that are configured in the IBM Spectrum Scale cluster.

The following capabilities are supported:

- The file-access capability (Unified file and object access)
- The multi-region capability (Multi-region object deployment)
- The S3 capability (Amazon S3 API)

If unified file and object access is configured, you can change the file-access capability to enable or disable the related services. Use the `mmobj file-access` command to change unified file and object access.

Use the `mmobj multiregion` command to change multi-region capabilities. Use the `mmobj s3` command to change S3 capabilities.

You can create a storage policy with unified file and object access only when file-access (unified file and object access) capability is enabled.

The **ibmobjectizer** and **openstack-swift-object-sof** services are started only if the **file-access** capability is enabled. Disabling the **file-access** capability stops these services.

For information about enabling, disabling, or listing object capabilities, see *Managing object capabilities* in *IBM Spectrum Scale: Administration Guide*.

Secure communication between the proxy server and other backend servers

Use this feature to establish secure communication between the proxy server and the backend Object Storage servers.

By default, object-server, object-server-sof, container-server, and account-server do not have authentication for the requests that they are serving. Processes, including the proxy-server that are connecting to these servers over their listening ports, can send requests that can result into updating the database and altering the object data on disk. Extra security between these servers can be enabled. Requesting process signs a request with a secret key kept in `swift.conf`. This key is verified by the serving object, container, or account server. To enable this feature, set:

```
mmobj config change --ccrfile swift.conf --section node_communication --property secure --value true
```

The signing middleware is added to proxy-server and the validating middleware is added to object-server, object-server-sof, container-server, and account-server. If the secret key is not present in `swift.conf`, it is randomly chosen and set to `key_secure_communication_secret` under `node_communication` section. In a multi-region environment, this key must be reset and kept common in all the clusters.

To revert to the original configuration, set:

```
mmobj config change --ccrfile swift.conf --section node_communication --property secure --value false
```

Note: Disable SSH access on the protocol nodes on the IBM Spectrum Scale cluster for the users that have the same UID and GID as the local swift user.

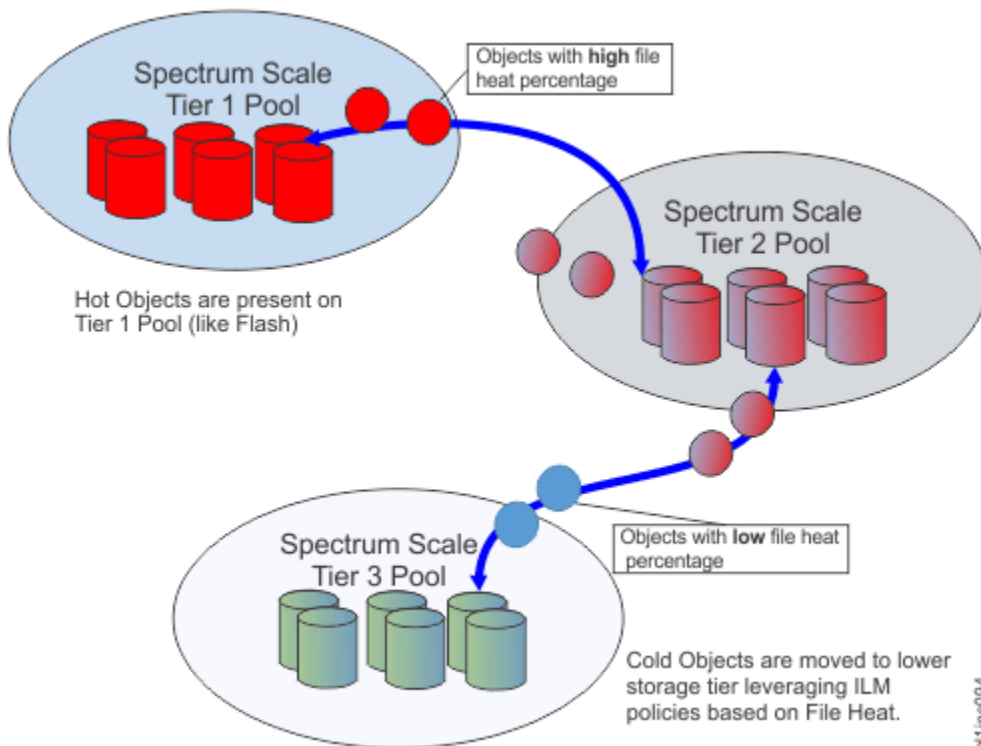
Object heatmap data tiering

Object heatmap data tiering policies can be applied to data that is frequently accessed.

For more information, see the topic *File heat: Tracking file access temperature* in the *IBM Spectrum Scale: Administration Guide*.

With the object heatmap data tiering policies, frequently accessed objects are stored in higher-performing storage pools such as the SSD-based pool. The policy also stores objects that are not accessed frequently in slower disk-based storage pools.

Note: Data that is not required for regular access is moved from slower disk-based storage pools to tapes.



The object heatmap policy

The object heatmap policy can be applied to object data files in the file system.

The policy manages the placement of objects in the gold, silver, and system pools. The policy also specifies the maximum threshold so that the capacity in the storage tiers is not over-utilized.

Enabling heatmap tracking

Enable heatmap tracking on the IBM Spectrum Scale file system by running the following command:

```
mmchconfig fileheatperiodminutes=1440,fileheatlosspercent=10
mmchconfig: Command successfully completed
mmchconfig: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

Active File Management

Active File Management (AFM) is a feature of IBM Spectrum Scale. You can use AFM to share data across clusters.

Introduction to Active File Management (AFM)

Active File Management (AFM) enables sharing of data across clusters, even if the networks are unreliable or have high latency.

The following figure is a sample of an AFM relationship.

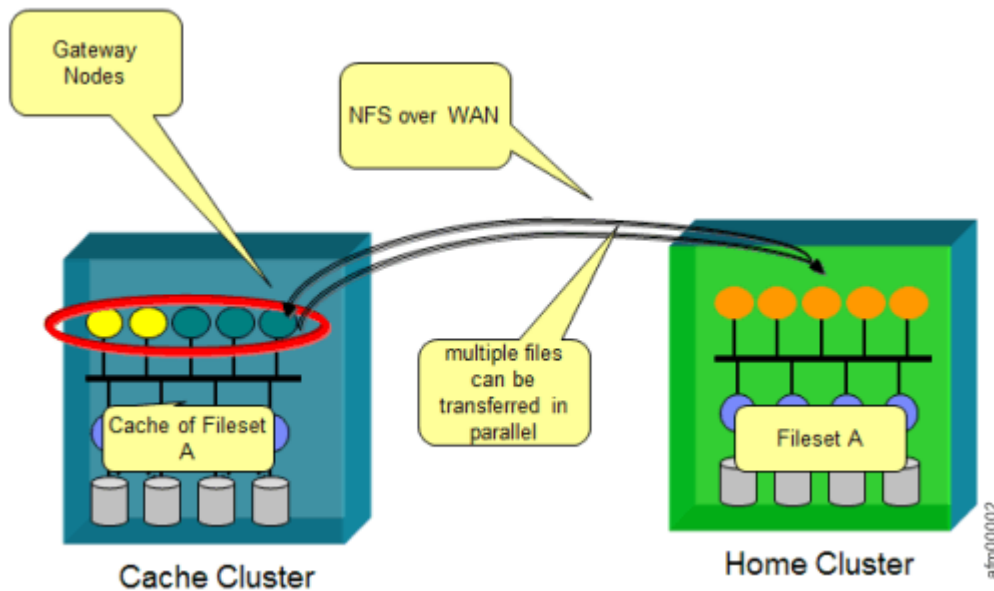


Figure 5. Sample of an AFM relationship

You can use AFM to create associations between IBM Spectrum Scale clusters or between IBM Spectrum Scale clusters and NFS data source. With AFM, you can implement a single namespace view across sites around the world by making your global namespace truly global.

By using AFM, you can build a common namespace across locations, and automate the flow of file data. You can duplicate data for disaster recovery purposes without suffering from wide area network (WAN) latencies.

AFM can be enabled on GPFS-independent filesets only. A dependent fileset can be linked under an AFM fileset, but only up to one level below the AFM-independent fileset. The dependent fileset does not allow nested dependent filesets under the AFM-independent fileset.

Individual files in the AFM filesets can be compressed. Compressing files saves disk space. For more information, see *File compression* in the *IBM Spectrum Scale: Administration Guide*.

Snapshot data migration is also supported. For more information, see *ILM for snapshots* in the *IBM Spectrum Scale: Administration Guide*.

Namespace replication with AFM occurs asynchronously so that applications can operate continuously on an AFM fileset without network bandwidth constraints.

Note: AFM does not offer any feature to check consistency of files across source and destination. However, after files are replicated, you can use any third-party utility to check consistency.

Overview and concepts

The following topics provide an overview of AFM and describe the concepts.

Cache and home

An AFM fileset can be enabled on a GPFS-independent fileset. Each fileset has a distinct set of AFM attributes. An IBM Spectrum Scale cluster that contains AFM filesets is called a cache cluster. A cache

cluster has a relationship with another remote site called the home, where either the cache or the home can be the data source or destination.

AFM constantly maintains an active relationship between the cache and the home. Changes are managed per fileset results in modular, scalable architecture capable of supporting billions of files and petabytes of data. Each AFM-enabled fileset is associated with a single home path.

AFM uses an NFSv3 or NSD (GPFS multi-cluster) protocol for the communication between the home and cache sites. A home export path is either an NFSv3 exported path or a multi-cluster or remote file system, which is mounted at the IBM Spectrum Scale cache cluster. This path is used by an AFM or AFM-DR fileset as a target path for data synchronization between sites. For AFM RO-mode filesets, the target path at the home NFS server can be exported as 'Read-Only' or 'Read/Write'. However, for AFM LU/SW/IW and AFM-DR mode filesets, the target NFS export path must be 'Read/Write'.

Example:

- Export as Read-Only for an AFM RO-mode fileset.

```
/ibm/gpfs0/homeDataSource GatewayIP/  
*(ro,nohide,insecure,no_subtree_check,sync,no_root_squash,fsid=1001)
```

- Export as Read/Write for an AFM-LU/SW/IW and AFM-DR mode fileset.

```
/ibm/gpfs0/homeDataSource GatewayIP/  
*(rw,nohide,insecure,no_subtree_check,sync,no_root_squash,fsid=1001)
```

Architecturally, AFM works with any NFS home export. However, if the home is a GPFS file system then AFM supports replicating ACLs, EA, and sparse files. You must issue the **mmafmconfig enable <ExportPath>** command on the home GPFS cluster to enable files replication.

Note:

1. If you do not run the **mmafmconfig enable** command and configure an AFM relationship, ACLs and extended attributes are not supported. Sparse file information is not maintained. The following error message appears in the `mmfs.log` file: `AFM: AFM is not enabled for fileset cachefs in filesystem gpfs. Some features will not be supported, see documentation for enabling AFM and unsupported features.`

Any cluster can be a home cluster, a cache cluster, or both. In typical setup, a home is in an IBM Spectrum Scale cluster and a cache is defined in another IBM Spectrum Scale cluster. Multiple AFM-enabled filesets can be defined in one cache cluster, and each cache has a relationship with targets with a home, or different cluster.

In IW, RO, and LU modes, multiple caches might point to the same home. But in SW mode, only one-to-one relationship between cache and home is supported. AFM can also be configured as a subscription service, where home is the data feed and all caches can subscribe to this data feed.

Within a single cache cluster, application nodes experience POSIX semantics. File locking across cache and home is not supported.

When you perform operations on AFM filesets, ensure that the operations are supported on home over the chosen protocol. Because the operations that are done from cache are replayed on the remote as normal file system operations. When you use the NSD protocol with UID remapping, operations such as `chown` (change ownership) are not supported.

Caching modes

The caching modes in AFM are Read-Only (RO), Single-Writer (SW), Local-Update (LU), and Independent-Writer (IW).

In a Read-Only mode, files in the cache fileset cannot be modified.

In a Single-Writer mode, cache pushes all changes to home but never checks home for any updates.

Independent-Writer mode allows both reads and writes, it pushes changes to home and checks home for file updates.

Local-Update is read only but files can be changed in the cache fileset, though the changes are never pushed to home and any changed file no longer checks home for updated versions.

Communication between cache and home

AFM uses the NFSv3 protocol or the NSD protocol for the communication between clusters.

Communication for caching between clusters is performed by one or more nodes on the cache cluster that are designated as gateway nodes by using the **mmchnode** command. The target path on the home server must be NFS-exported on one or more nodes in the home cluster, or the home file system must be mounted on the cache cluster by using the NSD protocol. For filesets in the AFM RO mode, the target path at home NFS export can be a read-only or read/write export. The exported path or the home-mounted path of the home cluster is used in the cache cluster as the target while creating a cache fileset.

You can export paths with Kerberos-enabled security levels from home to cache. At cache, configure Kerberos clients at gateway nodes and run the **mmchconfig afmEnableNFSec=yes -i** command. Gateway nodes can mount Kerberos-enabled home exports with security levels like `sys`, `krb5`, `krb5i`, `krb5p` after the **afmEnableNFSec** parameter is set to `yes` at the cluster level.

To configure Kerberos for AFM-DR filesets, the user must enable the read/write access to the AFM-DR secondary. The read/write access can be enabled by setting the **afmSecondaryRW** parameter value to 'yes'.

Run the following command at the AFM-DR secondary:

```
# mmchconfig afmSecondaryRW=yes -i
```

AFM maintains a relationship between cache and home by monitoring home availability and reconnects cache and home.

Note: If a home fileset is removed or deleted, you must remove the export path first and restart NFS at home.

The backend protocol - NFS versus NSD

The NSD protocol is a stateful protocol. The NFSv3 protocol is a stateless protocol, which is resilient to the low bandwidth and lossy networks.

The current recommended transport protocol for AFM data transfers is NFS due to the tolerance of NFS to unstable network connections. It is recommended to use NFS first and shift to the NSD protocol only if NFS does not meet the performance requirements even with multiple primary gateways and use of parallel data transfers. The implications of using the NSD protocol between the cache and home cluster are:

1. Network availability fluctuations and instability can affect the NSD protocol connection to the home on the cache cluster primary gateways. This network issue can lead to frequent data access interruptions from the home cluster, and can even cause the connection to the home cluster to stop responding. In these cases, it might be necessary to restart the GPFS daemon on the primary gateway, and possibly even restart the primary gateway server.
2. IBM Spectrum Scale instability issues on the home cluster can affect the cache cluster. The AFM fileset in the cache cluster does not respond because of the instability, and you might also need to restart the IBM Spectrum Scale service on the home and cache clusters.

For more information about setting up primary gateway nodes that communicate with multiple NFS servers at home, see [“Parallel data transfers” on page 59](#).

The following table summarizes the differences between NSD and NFS protocols on various parameters:

Table 6. Comparison between NSD and NFS protocols

	NSD	NFS
Usability	Customers are familiar with NSD use in multi-cluster environments. Configuration does not require NFS knowledge or tuning, but requires NSD tuning.	Configuration requires NFS knowledge and performance tuning for both NFS and TCP over WAN.
Performance	By default, uses all primary gateway nodes for parallel data transfers. Large file data transfer performance is better than NFS from a single primary gateway node as it can use the inherent parallelism of striping to multiple NSDs.	Parallel data transfers can be achieved by creating mapping between primary gateway nodes and NFS servers on the home. In summary, while both NFS and NSD can do similar forms of parallelism, generally NSD achieves higher performance.
Security	Encryption is built in, which can be turned on optionally.	Supports Kerberos-enabled exported paths from home to cache. The afmEnableNFSSec parameter must be set to yes on cache.
Firewall configuration	Special ports might not be required.	Must be configured for the traffic to pass through.
Stability	Performs well if the network is stable and has low latency.	More resilient to failures within the network such as packet drops that readily happen over WAN and it is more resilient, protecting the cache cluster from being affected by home cluster issues.

Considerations when you use the NSD protocol for AFM data transfers

The NSD protocol is more sensitive to packet drops and network latency than NFS. If the network does not respond, or if the packets are dropped, the NSD mount on cache cluster stops responding, causing the cache cluster also to stop responding. More causes of issues when the NSD protocol is used:

1. Deadlock in the home cluster - This deadlock might cause the NSD mounts on the cache cluster to stop responding for some time. Due to a non-responsive NSD mount, AFM fileset at cache that uses these NSD mounts as target might be in the 'unmounted' state. After the home cluster is responsive, the home cluster tries queued operations again.
2. Cluster reconfiguration or higher resource consumption on the home cluster - This configuration and resource consumption might cause a temporary loss of communication between home and cache cluster. If the home cluster does not respond within AFM wait timeout intervals, AFM filesets at cache that use these NSD mounts as target might be in the 'unmounted' state. After the home cluster is responsive, the home cluster tries queued operations again.
3. When a new primary gateway node joins the cluster, the old primary gateway node transfers the fileset to a new primary gateway node. If the remote file system is not mounted on the new primary gateway node, the fileset remains in the 'unmounted' state. After the remote file system is mounted at gateway node, the fileset automatically moves to the Active state.
4. Remote file system cannot be unmounted unless replication is stopped, or primary gateway node is restarted. AFM puts a hold on remote mount, not allowing the file system to be unmounted.
5. Creating an AFM association, by using GPFS protocol, to the same local file system is not supported.

Note:

- If the NSD mount on the gateway node is unresponsive, AFM does not synchronize data with home or secondary. The file system might be unmounted on the gateway node. A message AFM: Remote filesystem *remoteefs* is panicked due to unresponsive messages on fileset *<fileset_name>*, re-mount the filesystem after it becomes responsive. *mmcommon* *preunmount* invoked. File system: *fs1* Reason: *SGPanic* is written to *mmfs.log*. After the home or secondary is responsive, you must restore the NSD mount on the gateway node.
- When NSD backend is used, **afmDIO** parameter is set to 0 by default. For NFS backend, **afmDIO** parameter value must be set to 2.

Primary gateway and afmHashVersion

Each cache fileset in a cluster is served by one of the nodes that is designated as a gateway in the cluster. The gateway node that is mapped to a fileset is called the primary gateway of the fileset. The primary gateway acts as the owner of the fileset and communicates with the home cluster.

All other nodes in the cluster, including other gateways, become the application nodes of the fileset. Therefore, any node in the cache cluster can function as a gateway node and an application node for different filesets based on configuration of the node.

Application nodes communicate with the primary gateway for a fileset via internal network requests. The gateway function is highly available and can be scaled out. When a gateway node fails, all cache filesets that are owned by this gateway node are moved to another gateway node. This gateway node runs AFM recovery and takes over as the primary gateway of the filesets. When the old primary gateway returns after node failure, the original primary gateway resumes managing its fileset, and AFM transfers the queues for all the filesets from the current primary gateway to the old primary gateway.

The primary gateway can be configured to take help from other gateway nodes for parallel data movement of large files to and from home, if the home is an IBM Spectrum Scale cluster for increasing performance during data transfer. It is not recommended to assign the manager role to the gateway node because the assigned role can hamper the gateway node performance. The manager role can be assigned to other non-gateway nodes.

The gateway node designation is supported only on the Linux operating system. The Windows or AIX nodes cannot be designated as gateway.

The recommended setup configuration of a primary gateway node is as follows:

- The memory and CPU requirement for a gateway node depends on the number of assigned AFM or AFM-DR filesets and files inside the filesets. It is recommended to have minimum 128 GB of memory for the gateway node.
- Up to 20 AFM or AFM-DR filesets can be assigned to a gateway node.
- A gateway node is a dedicated node in the cluster without any other designations such as NSD server, CES protocol nodes, quorum, and manager.
- The recommended limit of the total number of inodes in all AFM, AFM-DR mode filesets that are assigned to the same gateway node is approximately 400 million.
- The following table contains the cluster level parameters:

Parameter	Value
Pagepool	8G
afmHardMemThreshold	40G
afmNumFlushThreads	8
afmDIO	2
maxFilestoCache	10000
afmMaxParallelRecoveries	3

To set these parameters, use the **mmchconfig** command. For example,

```
# mmchconfig pagepool=8G
```

For more information about configuration parameters, see *Parameters for performance tuning and optimization* in *IBM Spectrum Scale: Administration Guide*.

This configuration recommendation is based on observations in a controlled test environment by running moderate or reasonable workload. The parameters values might vary based on the setup or workload.

Note: AFM gateway node is licensed as a server node.

See “Parallel data transfers” on page 59 to set up gateway nodes that communicate to multiple NFS servers at home.

afmHashVersion

In a cluster with multiple gateway nodes and AFM cache filesets, AFM uses a hashing algorithm to elect the primary gateway for each of the filesets. This algorithm can be selected by changing the cluster level config tunable **afmHashVersion**. Issue the following command to change the hashing algorithm.

```
# mmchconfig afmHashVersion=version
```

Note: You need to shut down the cluster and then issue this command because you cannot set **afmHashVersion** on an active cluster.

AFM Hash Version 2

This version is the default version. The hashing algorithm is set by default on an IBM Spectrum Scale 4.1 cluster. This version is based on the static fileset ID mapping. On an upgraded cluster, the old hashing algorithm is effective by default. You can change the hashing version by changing the value of the **afmHashVersion** tunable.

AFM Hash Version 4

This version is an improved hash version '2'. If you have many filesets and if these filesets are not evenly distributed across gateway nodes, you can set the **afmHashVersion** parameter to 4 for the improvement. This hashing version is dynamic, which ensures balanced mapping of AFM filesets and gateway nodes.

Note: Do not set the **afmHashVersion** parameter to 4, if AFM has heavy data transfer operations from the gateway node to filesets. You are not recommended to set this version because you might face some deadlock issues in some cases. You can choose version '5' over version '4'.

AFM Hash Version 5

For better load balancing of AFM or AFM-DR filesets across gateway nodes, you can set the **afmHashVersion** parameter to '5'. This version is recommended for the load balancing of the filesets. AFM hash version '5' does not cause any known performance degradation in comparison with earlier versions. This option must set only at the cache or primary cluster and at the client cluster if it has remote mounts, that is, the AFM cache cluster. However, do not set this option at the home or secondary cluster because the home or primary cluster does not have AFM or AFM-DR filesets.

AFM hash version 5 supports manual assignment of the selected gateway node to an AFM or AFM-DR fileset by using the **mmafmctl stop** command and the **mmafmctl start** command. You can change the default assigned gateway node of an AFM or AFM-DR fileset by using the **mmchfileset** or **mmcrfileset** command. When the specified gateway node is down or not available, then AFM internally assigns a gateway node from the available gateway node list to the fileset. For filesets that do not have the **afmGateway** parameter set, the gateway node is assigned by using the hash version 2.

To assign the **afmGateway** parameter to a fileset, see the *mmchfileset* command or *mmcrfileset* command in the *IBM Spectrum Scale: Command and Programming Reference*.

You must stop replication on an AFM or AFM-DR fileset before you set the **afmGateway** parameter and start replication after the parameter is set.

Do the following steps to assign a gateway node by using the **mmchfileset** command:

1. Stop replication on an AFM or AFM-DR fileset by issuing the following command:

```
# mmafmctl fs stop -j AFM/AFM-DR Fileset
```

2. Set the **afmGateway** parameter by issuing the following command:

```
# mmchfileset fs AFM/AFM-DR Fileset -p afmGateway=NewGatewayNode
```

3. Start replication on the AFM or AFM-DR fileset by issuing the following command:

```
# mmafmctl fs start -j AFM/AFM-DR Fileset
```

To assign a gateway node by using the **mmcrfileset** command, issue the following command:

```
# mmcrfileset fs fileset -p afmMode=mode,afmTarget=Target,afmGateway=Gateway --inode-space new
```

For more information about start and stop replication on a fileset, see [“Stop and start replication on a fileset”](#) on page 90.

Note:

- To set the **afmHashVersion** parameter, IBM Spectrum Scale cluster must be shut down and start after this cluster level tunable is set for a new hash algorithm to take effect.
- To change the **afmHashVersion** parameter to ‘5’, all the nodes in the AFM cache and the client cluster must be upgraded to the minimum 5.0.2 level.

Global namespace

You can combine the home and cache entities to create a global namespace.

Any client node can use the same path to connect to the data within any of the IBM Spectrum Scale clusters that are part of the namespace.

In such a global namespace, the following AFM features can improve application performance on a remote site:

1. When a file is being read into a cache, the data can be read after it arrives in the cache.
2. Multiple cache filesets can share a single file system. Data can be transferred between sites in parallel.

AFM also performs on networks that are unreliable, or have high latency. The following example is of a global namespace, which implemented by using AFM, with three different sites. An IBM Spectrum Scale client node from any site sees all of the data from all of the sites. Each site has a single file system. Each site is the home for two of the sub directories and cache filesets that point to the data that originates at the other sites. Every node in all three clusters has direct access to the global namespace.

The following figure shows global namespace that is implemented by using AFM.

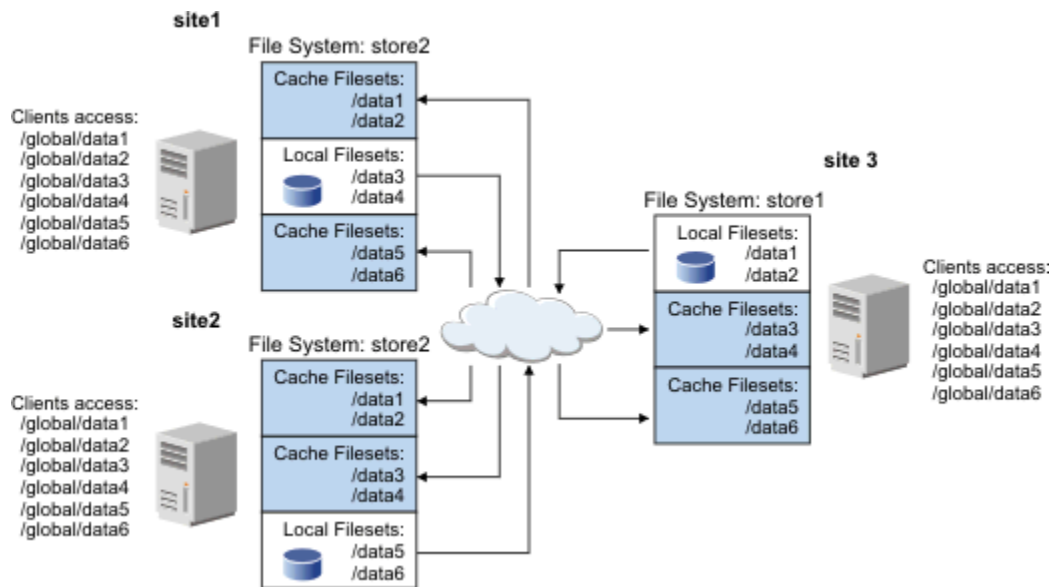


Figure 6. Global namespace implemented by using AFM

Revalidation

As users traverse the directory tree of an AFM cache fileset, files and directory metadata information from the home cluster is checked and updated as needed on the cache cluster. This process is called AFM revalidation.

Revalidation performance is dependent upon network latency and bandwidth available between the cache and the home. Revalidations are done per node, and not per fileset. If a file or directory is revalidated from one node on the cache cluster, the same fileset goes through another revalidation operation when accessed from another node on the same cache cluster. You can modify the refresh intervals by using the following command:

```
# mmchfileset fs1 sw1 -p afmFileOpenRefreshInterval=160
```

In this example, the **afmFileOpenRefreshInterval** parameter is set to 160 for the **sw1** fileset in the **fs1** file system.

Revalidation intervals can be adjusted to support the workload and network latency. Setting a parameter by using the **mmchconfig** command sets the default for all filesets. Parameters set by using the **mmchfileset** command affect only a particular fileset and override the global defaults. You can enable, modify, or disable any of the intervals based on the application needs, though it is recommended to use default values for most cases.

If file or directory refresh intervals are disabled for a fileset, the refresh intervals can be enabled by using the **mmchfileset** command. Enabling requires the fileset to be unlinked, and then linked again.

For more information, see the *mmchfileset* command in the *IBM Spectrum Scale: Command and Programming Reference*.

It is recommended not to set the revalidation intervals to 0, as a revalidation request is continuously sent to home, thus resulting in performance degradation. You must set the revalidation interval to as large as possible, depending on how frequently home gets updated, and at what interval the cache needs the updated data.

For more information about revalidation intervals, see the *mmcrfileset* command in the *IBM Spectrum Scale: Command and Programming Reference*.

The revalidation intervals are defined by the following configuration parameters. These parameters are tunable at the cluster and fileset level and can be changed by using the **mmchconfig** command and the **mmchfileset** command at the cluster and the file level:

1. **afmFileLookupRefreshInterval**: The frequency of revalidation that is triggered by a lookup operation on a file such as ls or stat, from the cache.
2. **afmDirLookupRefreshInterval**: The frequency of revalidation that is triggered by a lookup operation on a directory such as ls or stat, in the cache.
3. **afmFileOpenRefreshInterval**: The frequency of revalidations that are triggered by read or write operations on a file in the cache. Open requests on that file are served from the cache fileset until the **afmFileOpenRefreshInterval** expires after which the open requests are sent to home.
4. **afmDirOpenRefreshInterval**: The frequency of revalidations that are triggered by read or update operations on a directory from the cache. 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 home.
5. **afmRevalOpWaitTimeout**: The time for which AFM waits for completion of revalidation to get response from the home cluster. The **afmRevalOpWaitTimeout** parameter can be set only for a cluster and not for an individual fileset.

RO, LU, and IW filesets revalidate regularly with home. The SW mode populates metadata only one time during first access and does not revalidate with home thereafter. To revalidate, AFM sends a message to the home cluster to find out whether the metadata of that file or directory is modified since the last time it was revalidated. If so, the latest file metadata or data information, depending on the type of request, at home is reflected on the cache.

Revalidation in asynchronous mode

Synchronization with home before revalidation delays response to the applications querying data. To overcome this delay, you can perform cache data refresh operation in asynchronous mode. You must set the **afmRefreshAsync** parameter to 'yes'.

For more information about the **afmRefreshAsync** parameter, see *mmchconfig* command in the *IBM Spectrum Scale: Command and Programming Reference* and *Configuration parameters for AFM* in the *IBM Spectrum Scale: Administration Guide*.

Cached and uncached files

A `readdir` operation on a directory populates the metadata of the directory in the cache, but this operation does not populate contents of each file within the directory. A read operation on file generates a request to the home to make contents available in the cache. The file contents do not need to be in the cache to start reading it.

AFM allows data to be pre-populated before actual read operation by using the **mmafmctl prefetch** command. For more information about pre-populating data, see [“Prefetch” on page 63](#).

A file whose contents are available in the cache is called a cached file. A file whose contents are not yet present in the cache is called an uncached file. An uncached file cannot be evicted, resynched with home, or failed over to a new home. See the sections on Asynchronous operations and delay, for AFM eviction and syncing to home.

Files that have all blocks read, or the entire file contents that is fetched, are marked as cached. AFM does whole-file caching by default. By default, reading more than three blocks of a file drives AFM to cache the full file in the background for performance. Sometimes the whole file might not need to be cached. For example, some applications read only a few bytes of a file to detect the file mime type. In such cases, you can configure partial file read behavior on cache fileset. For more information about partial file caching, see [“Partial file caching” on page 63](#).

When a write operation is performed directly on an uncached file in IW, SW, or LU modes, AFM fetches first the entire file to the cache (cached) by default and later allows the local write to proceed. This behavior is for in-place writes within the size of the file. If the append that is being performed to the file is more than the last offset, the behavior is slightly different:

- The append on files in IW or SW filesets appends only data to the file at the cache. This append can leave the rest of the file data from the home as uncached.

- The append on files in the LU mode must mark the file as dirty, and the append disconnects the appended file with an appended file of the cache. So that AFM fetches the entire file even for an append more than the file size.

Likewise, the files that originate in the cache on IW, SW, or LU filesets are marked as cached by default because these files are created and written to at the cache.

Synchronous or asynchronous operations

AFM operations are serviced either synchronously or asynchronously. Reads and revalidations are synchronous operations, and update operations from the cache are asynchronous.

Synchronous operations require an application request to be blocked until the operation completes at the home cluster. File data is available while AFM queues are processed asynchronously in the background.

If synchronous operations depend on the results of one or more updates or asynchronous operations, AFM prioritizes the dependent asynchronous operations before the execution of the synchronous operations. For example, synchronous operations like file lookup cause dependent asynchronous operations to be flushed, overriding asynchronous delay (**afmAsyncDelay**).

AFM deploys a filtering algorithm for most optimal flushing performance by analyzing the queued requests. For example, if a write and a delete are in a queue on the same file, the write request is dropped from the queue and the delete is run at home. Similarly, when **mkdir** and **rmdir** of the same directory name are in a queue, both requests are dropped.

AFM sends data from cache to home as root. If home has fileset quotas at home, it must be set such that it can accommodate all writes from cache.

The **afmRefreshAsync** parameter changes the synchronous behavior. This parameter causes a **readdir**, which must be a synchronous command, to behave asynchronously. That is, like a follow-up **readdir** for an already cached directory. The **Async readdir** goes in the queue like any other asynchronous operation, where the cache serves existing **readdir** contents locally.

This **afmRefreshAsync** is useful for IW, RO, and non-dirty or unchanged LU modes that need to perform revalidations frequently with the home. The applications must not be affected with such a **readdir** on the application path.

Asynchronous delay

All update operations from the writable cache filesets are on the primary gateway. Queues in the primary gateway are pushed to home asynchronously based on the **afmAsyncDelay** interval.

This interval can be modified by using the **mmchfileset** command. You can force a flush of all the pending updates without waiting for asynchronous delay by using the **mmafmctl** command with the **flushPending** option. The **flushPending** option can be set at the fileset or file system level, or with a file that contains the list of files to be flushed.

For more information, see **mmafmctl** command in *IBM Spectrum Scale: Command and Programming Reference*.

Operations with AFM modes

Following are the operations with AFM modes - Read only (RO), Single writer (SW), Local updates (LU) and Independent writer (IW).

1. Read only (RO)

The following figure illustrates the Read Only mode.

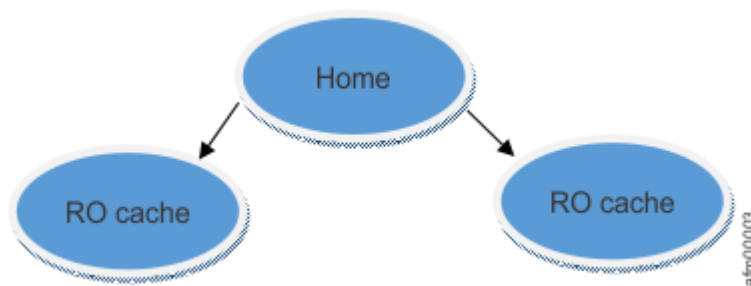


Figure 7. Read only mode

In this mode, data in the cache is read-only. Creating or modifying files in the cache fileset is not allowed. If a file is renamed at the home for an RO fileset, the file is re-created in cache and is assigned a new inode number in the cache. If the file is in use by an application while it is re-created (deleted and re-created with the same name) at home, it gets re-created in cache.

2. Single writer (SW)

The following figure illustrates the Single writer mode.



Figure 8. Single writer mode

In this mode, only the cache fileset does all the writing and the cache does not check home for file or directory updates. The administrator needs to guarantee no writes are done in the home cluster. AFM does not enforce this check.

An SW home can have some pre-existing data. An SW cache can cache and update this data. Update of an uncached file from SW home caches the file. However, the truncate or append operations on an uncached file does not fetch the contents of the file into the cache, but queues the truncate or append operation to the home.

3. Local updates (LU)

Local update is similar to read-only mode although you can create and modify files in the cache fileset. Updates in the cache are considered local to the cache and get decoupled from the corresponding file at home. Local updates are never pushed back to home. After modifications in a file, the file is no longer compared to the version at home during revalidation to verify that it is up to date. Changes of this file on home do not have an impact on the cached copy of the file and vice versa.

Behaviors with local files: In AFM, LU mode files have one of the following states:

- **Uncached:** Files on the home for which metadata but no data is copied into the cache are shown in the cache as uncached. The file is not resident on cache, but only on the home. Changes on the home are reflected in the cache.
- **Cached:** If an uncached file is read in the cache or pre-fetched, the state of the file changes to cached. In the cached state, all changes to the file on the home are reflected in the cache. The file is resident on the cache.
- **Local:** File data or metadata that is modified at cache becomes local to the cache. The cached files relationship to the file in the home is broken. Changes on the home are not reflected in the cache anymore and file changes are not copied to the home.

Operations in the cache that trigger the transitions between these states are shown below:

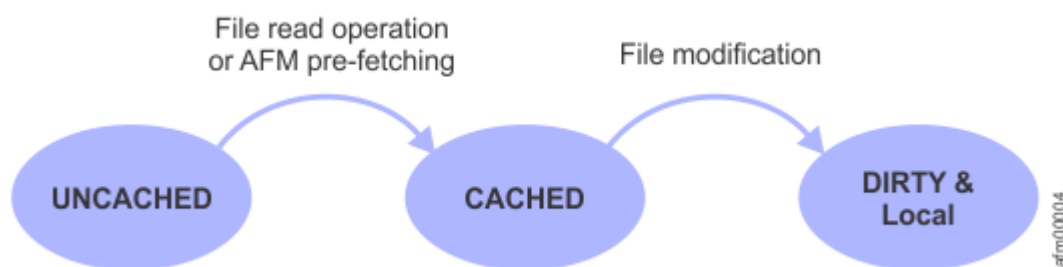


Figure 9. Behaviors with local files

The following tables summarize AFM LU mode behavior with local files:

File operation at home	Uncached file	Cached file	Local file
New Create	Reflects	Reflects	Reflects
Change data or attributes/Rename/Delete	Reflects	Reflects	Does not reflect

File operation at home	Uncached file	Cached file	Local file
File lookup revalidation	Pulls the metadata	Pulls the latest metadata	Does not pull the latest metadata
Read	Pulls the data	Pulls the latest data	Shows data in the local file – does not pull the latest data from the home
Prefetch	Prefetches the data	Prefetches the latest data	Does not prefetch the latest data from home
Change data	Pulls the data into cache and updates the changes	Pulls the latest to cache and updates the changes	Updates the local copy of the file and does not pull the latest data from the home before the update
Change attributes/metadata except delete EA	Updates the file and marks file as local	Updates the file and marks file as local	Updates the file – does not pull the latest metadata from the home before the update

Directories in a local-update cache:

Directories can become local in the LU mode with the following directory operations from the cache:

- Deleting files or sub-directories in the cache directory
- Creating new files or sub-directories in the cache directory
- Renaming files or sub-directories in the cache directory

The following file operations on cache do not cause directories on the cache to become local:

- Update the uncached or cached file
- Add attributes to the file
- Migrate the cached or local files to tape
- Recall of the cached or local files from tape

A local directory in an LU cache can contain local, cached, and uncached files and directories. A local directory is not revalidated with home. Therefore, operations on the directory at home do not show up on the cache.

The following table summarizes AFM LU mode behavior with local directories:

Operation at home	Cache behavior - Normal directory	Cache behavior - Local directory
Create a file or subdirectory	Reflects in cache after next directory lookup	Does not show in cache after next directory lookup. Contents can be read or prefetched if the file name is known in the cache. After read or prefetch, the new file or directory shows up in the cache for future file lookup
Update data or attributes of an existing file or subdirectory	Reflects in cache after next file lookup	Reflects in cache after next file lookup
Rename an existing file or subdirectory	Reflects in cache after next file lookup	Does not show in cache after next directory lookup. On reading the file, contents from renamed file that is fetched into the cache into the old file name, old file name always prevails in the cache. New file name can arrive at the cache with prefetch, in which case both names will co-exist and point to the same inode at home.
Delete an existing file or subdirectory	Reflects in the cache after next directory lookup	Deleted upon the next file lookup

Directory operation from cache	Cache behavior - Normal directory	Cache behavior - Local directory
Directory revalidation	Pulls the latest metadata	Does not pull the latest metadata
Read a non-local file in the directory	Pulls the latest data	Pulls the latest data
Prefetch a non-local file	Prefetches the latest data	Prefetches the latest data
Create a file or directory	Updates and marks the directory local	Updates the local directory
Change data of a non-local file	Pulls the data into the cache, updates the file and marks file as local	Pulls the data into the cache, updates the file and marks the file as local
Change metadata of a non-local file except delete of EA	Updates the file and marks the file as local	Updates the file and marks the file as local

Appending to, truncating, or writing to an uncached file in LU mode fetches the entire file to the cache before making the change locally.

If you expect revalidation, change the LU fileset root directory with caution as this might cause the fileset to be marked as local, and context with the home is lost. For example, running **chmod** or **chown**

of the LU fileset root directory causes all sub directories from fileset root to be out of synchronization with the home.

4. Independent writer (IW)

The following figure illustrates the Independent writer mode.

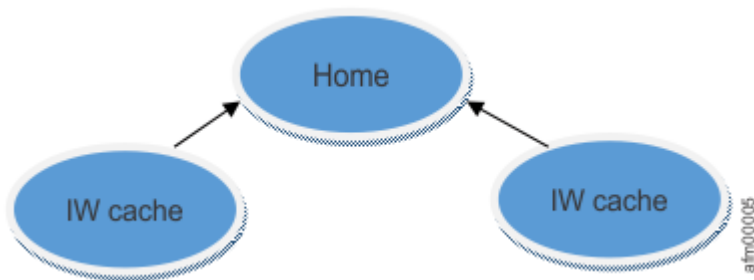


Figure 10. Independent writer mode

This mode allows multiple cache filesets to point to the same home. There is no synchronous locking between clusters while updating home. Each cache reads from home and makes updates to the home independently of each other, based on the revalidation intervals and asynchronous delay.

This mode is used to access different files from each IW cache site, such as, the unique users at each site updating files in their home directory. While this mode allows multiple cache clusters to modify the same set of files, this should be only done by advanced users. This is because there is no locking or ordering between updates. Updates are propagated to the home in an asynchronous manner and can be delayed due to network disconnections. Therefore, conflicting updates from multiple cache sites can cause the data at the home site to be undetermined.

Writing in-place on a pre-existing uncached file at the home pulls the complete contents of the file into cache. However, truncate and append operations on an uncached file do not fetch the contents of the file into cache, but queues the truncate and append operations to home.

When many IW cache filesets point to the same NFS home, the number of NFS threads at home can be tuned for better efficiency. Increasing the revalidation intervals for the IW cache filesets might reduce the frequency of periodic refreshes from home and improve cache performance.

Note: Do not create hard links at home or in the cache when IW is used, as IW failback does not preserve hard links.

The following file attributes are maintained locally in cache: Control attributes, direct I/O, replication factors, fileset quotas, storage pool flags, special file types such as FIFO, socket, block, character device. Hard links can be created at cache. Hard links at home are displayed as individual files in cache.

Filesets to the same home target

The following are the possibilities of filesets that point to home as the target:

- One or more RO filesets can point to the same target.
- One or more LU filesets can point to the same target.
- One or more RO and LU filesets can point to the same target.
- SW, RO/LU filesets can point to the same target.
- Only one SW and one or more RO or LU filesets can point to the same target.
- One or more IW, RO, and LU filesets can point to the same target.
- More than one SW fileset must not point to the same target. More than one SW fileset can technically point to the same target but can be only a writer. The other must behave like a reader fileset.
- One or more IW filesets can point to the same target.
- SW and IW filesets must not point to the same target.

Conversion of mode

An Active File Management (AFM) cache can be converted to another mode in some conditions to meet all types of requirements of data management by unlinking the fileset and by using **mmchfileset -p afmmode** command.

The mode conversion limitations are as follows:

- A single-writer (SW) or independent-writer (IW) fileset with pending requests in the queue cannot be converted.
- Local-update (LU) filesets cannot be converted to any other mode.
- Before you convert a fileset from the IW mode to the SW mode, ensure that all the remaining IW filesets are converted to either RO or LU. This conversion is necessary to avoid home conflicts that are caused by writers from other IW filesets.
- You cannot change the mode or disable AFM when the fileset is linked.

Internal AFM directories

AFM uses `.afm`, `.ptrash`, `.pconflicts`, and `.afm/.afmtrash` as internal directories.

AFM uses the following internal directories.

Important: Do not alter or remove any of these directories.

1. The `.afm` directory: Present in the cache and the home.

The **mmafmconfig enable** command creates this directory in the home cluster. The **mmafmconfig disable** command removes the directory from home. The `.afm` directory is created in the cache when you create a fileset, and is valid throughout the lifetime of the cache fileset.

2. The `.ptrash` directory: Present in the cache.

The existing `.ptrash` directory is applicable to the RO and LU modes.

For these modes, when a directory along with files or directories is moved to a cache fileset, and the same directory is removed from the home fileset, then the cache fileset directory is moved to the `.ptrash` directory.

3. The `.pconflicts` directory: Present in the cache.

4. The `.afm/.afmtrash` directory: Present in the home.

This directory is created at the home when the **mmafmconfig enable** command is run, or AFM creates it automatically from the cache after the relationship is initialized. Directories that are conflicting at the home are moved to this cache directory to resume replication successfully.

If an SW or IW mode fileset has deletion conflict for a file, the conflict is resolved by removing the file from the cache and moving it to `.ptrash`. A conflict occurs when a file in the home is removed but still exists in the cache because the cached version has outstanding changes. A file with outstanding changes in the cache not yet copied to home is called dirty. Files that are moved from the `.ptrash` or `.pconflicts` directory by using the **mv** command are treated as local files to the cache fileset and are not queued to the home. However, if these files are copied, for example, by using the **cp** command to the cache, they are queued to the home as new files.

Active File Management (AFM) features

The following sections list features of Active File Management (AFM).

AFM Network File System version 4 support

Network File System version 4 (NFSv4) is a stateful protocol. AFM uses an NFS client for replication. The client on an AFM gateway node can mount the exports by using either NFSv3 or NFSv4. AFM does not differentiate between an actual mount version except where ACLs from a third-party file system are migrated by using NFSv4. NFSv4 is more secure and improves the replication performance even on a high latency network. NFSv3 is not secure enough to use through firewalls.

By setting the **afmNFSTVersion** parameter on a cluster, you can enable NFSv4 for the communication between the home and the cache. The default value of this parameter is 3. The allowed values for the **afmNFSTVersion** parameter are 3, 4.1 and 4.2 for the kernel NFS server and 3 and 4.1 for the Ganesha NFS server. The NFS version can be changed by setting the **afmNFSTVersion** value. For example, to change NFSv4 to NFSv3, use the **mmchconfig afmNFSTVersion = 3 -i** command. This parameter can be set only at the cluster level. Thus, filesets can either use NFSv3 or NFSv4. Both versions cannot be run simultaneously in the cluster. For more information about the **afmNFSTVersion** parameter, see *mmchconfig command* in *IBM Spectrum Scale: Command and Programming Reference*. For more information, see *Enabling AFM NFSv4* in *IBM Spectrum Scale: Administration Guide*.

With the NFSv4 support, AFM can fetch the NFSv4 ACLs from third-party file server to an IBM Spectrum Scale AFM fileset. For more information about the **afmSyncNFSv4ACL** parameter, see *mmchconfig command* in *IBM Spectrum Scale: Command and Programming Reference*.

NFSv4 ACL conversion examples

1. Display ACL that is set on an external file system on the home.

```
# getfacl /ext4/dir1/1.txt
```

A sample output is as follows:

```
getfacl: Removing leading '/' from absolute path names
# file: ext4/dir1/1.txt
# owner: root
# group: root
user::rw-
user:user12:rw-
group::r--
group:user12:rw-
mask::rw-
other::r--
```

2. A single writer AFM mode fileset is created and data is cached. Check the directory contents.

```
# cd /gpfs/gpfs1/sw1
```

```
# ls -l
```

A sample output is as follows:

```
total 0
-rw-rw-r-- 1 root root 3 Apr
-rw-rw-r-- 1 root root 3 Apr
8 15:08 1.txt
8 15:08 2.txt
```

3. Display NFSv4 ACL on the cache by issuing the **getfacl** command.

```
# getfacl 1.txt
```

A sample output is as follows:

```
# file: 1.txt
# owner: root
# group: root
user::rw-
user:user12:rw-
group::r--
mask::rw-
group:user12:rw-
other::r--
```

4. Display NFSv4 ACL on the cache by issuing the **mmgetacl** command.

```
# mmgetacl -k nfs4 1.txt
```

A sample output is as follows:

```

#NFSv4 ACL
#owner:root
#group:root
special:owner@:--x--:deny
(-)READ/LIST (-)WRITE/CREATE (-)APPEND/MKDIR (-)SYNCHRONIZE (-)READ_ACL (-)READ_ATTR
(-)READ_NAMED
(-)DELETE
(X)DELETE_CHILD (-)CHOWN
(X)EXEC/SEARCH (-)WRITE_ACL (-)WRITE_ATTR
(-)WRITE_NAMED
special:owner@:rw-c: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
(-)EXEC/SEARCH (X)WRITE_ACL (X)WRITE_ATTR
(X)WRITE_NAMED
user:user12:rw-x-:allow
(X)READ/LIST (X)WRITE/CREATE (X)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
group:user12:rw-x-:allow
(X)READ/LIST (X)WRITE/CREATE (X)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:group@:r---:allow
(X)READ/LIST (-)WRITE/CREATE (-)APPEND/MKDIR (X)SYNCHRONIZE (X)READ_ACL (X)READ_ATTR
(X)READ_NAMED
(-)DELETE
(-)DELETE_CHILD (-)CHOWN
(-)EXEC/SEARCH (-)WRITE_ACL (-)WRITE_ATTR
(-)WRITE_NAMED
special:everyone@:r---:allow
(X)READ/LIST (-)WRITE/CREATE (-)APPEND/MKDIR (X)SYNCHRONIZE (X)READ_ACL (X)READ_ATTR
(X)READ_NAMED
(-)DELETE
(-)DELETE_CHILD (-)CHOWN
(-)EXEC/SEARCH (-)WRITE_ACL (-)WRITE_ATTR
(-)WRITE_NAMED

```

Data migration by using Active File Management

IBM Spectrum Scale Active File Management (AFM) supports migration of data from any legacy appliances or old GPFS system to the latest IBM Spectrum Scale cluster.

The migration is required when you replace an old hardware setup or upgrade the hardware. Data from source can be migrated by using either 'nfs' or 'gpfs' (NSD multi-cluster) based protocol. The data migration can be enabled either at the independent filesset-level or at the file system-level. For the migration, AFM RO (Read-Only) mode and AFM LU (Local-Update) mode filessets are supported.

AFM supports the following configurations for the data migration:

1. Migration of data from a source node or a cluster to an IBM Spectrum Scale AFM filesset.
2. Migration of data from an old file system to an IBM Spectrum Scale AFM filesset on the same cluster.
3. Migration of data from a legacy source or an old file system to an AFM enabled IBM Spectrum Scale file system.

The migration procedure is the same for all these configurations. When the migration is performed at the filesset level, an AFM RO mode filesset is created first and then the AFM RO mode is converted into an AFM LU-mode filesset. After the migration is completed, the AFM LU-mode filesset might be converted into a GPFS-independent filesset by disabling the AFM relationship.

When the data migration is planned at the file system level, AFM can be enabled by providing AFM-specific parameters at the time of file system creation. For the file system-level migration, no new filesset creation is required. An AFM RO mode or AFM LU-mode filesset is enabled on the default filesset of a file system, that is, on a root filesset. After the migration is completed, the AFM relationship might be disabled.

After the AFM relationship is disabled from a GPFS file system, this file system becomes a regular GPFS files system. If you disable the AFM relationship, you cannot enable it.

The migration procedure is the same for the file system-level migration and the fileset-level migration. For more information about the migration, see *Data migration by using AFM migration enhancements* in the Library and related publications.

To improve performance of the file system-level migration by using AFM, a new AFM tuning parameter must be enabled at the root fileset of the file system when you enable AFM, that is, **afmGateway=all**. If you set, this parameter allows AFM to queue operations based on file or inode to a gateway node. This parameter must be enabled for the file system-level migration use case. To enable this parameter, a cluster-level parameter **afmHashVersion** value must be set to '5'. This parameter works only with hash version 5 and it is not effective on less than this version. For the **afmGateway=all** parameter, the file system level must be 5.0.2 or later and the minimum release level is 5.1. For file system-level migration, all nodes must be at 5.1 release level.

This parameter can be enabled either at the time of file system creation or by using **start/stop** command.

The following examples show how to enable file system-level migration by using AFM:

1. To enable AFM data migration at file system level, issue the following command:

```
# mmcrfs fs1 "disk1;disk2;disk3;disk4" -k all -Q yes -T /gpfs/fs2 --perfilesset-quota -p  
afmMode=ro,  
afmTarget=:/<Source_Path>,afmGateway=all
```

2. To enable **afmGateway=all** after the creation of file system, issue the following commands:

```
# mmafmctl fs1 stop -j root  
# mmchfilesset fs1 root -p afmGateway=all  
# mmafmctl fs1 start -j root
```

Fast create

When the fast create feature is enabled on AFM or AFM-DR filesets, this feature reduces the number of operations that are generated at the time of file creation at the cache. AFM needs to replicate each file creation operation to the home as a separate operation.

Every time when a file is created, operations such as create, truncate, write, chmod, chown, or set ACL/EA are generated at the file system level. Each operation is replicated to the target site as a separate operation. The applications that generate such workload are tar, git, or make, which increases stress on the gateway node to queue all file creation operations as a separate operation for each file creation. Each operation that is performed from an application node for a file creation triggers a separate RPC send or acknowledge to or from the gateway node.

On the clusters where the fast create feature is enabled, AFM does not send all operations. However, AFM combines all operations of a file creation and sends these combined operations to the target. The target performs all operations that are required so that both sites are in sync. This feature helps you to avoid queuing all operations and send only one operation for every file creation. Thus, the fast create feature improves the performance by saving many RPC exchanges between the application node and the gateway node and also saves memory and network bandwidth.

For the performance improvement, the **afmAsyncDelay** parameter must be tuned when the **afmFastCreate** parameter is set on an AFM or AFM-DR fileset. The tuning of the **afmAsyncDelay** parameter depends on the application and network workload. If you set the **afmAsyncDelay** parameter value too high, more memory is consumed on the gateway node. To change the value of the **afmAsyncDelay** parameter, issue the following command:

```
# mmchfilesset fs -j filesset -p afmAsyncDelay=Value
```

This feature can be enabled or disabled on any individual AFM or AFM-DR fileset. To enable this feature, all the nodes in the cluster must be upgraded to the latest release level and the latest file system format level. Before you enable this feature on a fileset, you must stop the fileset and then set this

parameter. After you enable this feature, you need to start the fileset to continue the replication. For more information about stop and start the fileset, see [“Stop and start replication on a fileset”](#) on page 90.

Do the following steps to enable this feature on an AFM or AFM-DR fileset:

1. Update the cluster release level to the latest level and the file system version to the latest version. For more information about the cluster release level, see [“Completing the upgrade to a new level of IBM Spectrum Scale”](#) on page 572, and for more information about file system version, see *File system format changes between versions of IBM Spectrum Scale* in the *IBM Spectrum Scale: Administration Guide*.
2. Ensure that the fileset state is active and operations not are in the queue. To check the fileset state, issue following command:

```
# mmafmctl fs getstate -j fileset
```

3. To stop the fileset, issue the following command:

```
# mmafmctl fs stop -j fileset
```

Note: If you set the **afmFastCreate** parameter without stopping or unlinking a fileset,

```
# mmchfileset fs fileset -p afmFastCreate=yes
```

the following error is displayed:

```
mmchfileset: [E] Fileset cannot be changed, either fileset is linked or not stopped.  
mmchfileset: Command failed. Examine previous error messages to determine cause.
```

4. To enable or disable the fast create at the fileset level, issue the following command:

```
# mmchfileset fS -j fileset -p afmFastCreate=yes|no
```

5. Start the fileset by issuing the following command:

```
# mmafmctl fs start -j fileset
```

6. Ensure that the fast create is enabled successfully by issuing the following command:

```
# mmlsfileset fs fileset -L --afm
```

For example,

```
Filesets in file system '<FS>':  
  
Attributes for fileset <Fileset>:  
=====
```

Status	Linked
Path	<Fileset_Linked_Path>
Id	236
Root inode	30932995
Parent Id	0
Created	Tue Mar 1 07:35:21 2020
Comment	
Inode space	59
Maximum number of inodes	100096
Allocated inodes	100096
Permission change flag	chmodAndSetacl
afm-associated	Yes
Target	nfs://<home>/<Fileset_Linked_Path>
Mode	single-writer
File Lookup Refresh Interval	30 (default)
File Open Refresh Interval	60 (default)
Dir Lookup Refresh Interval	60 (default)
Dir Open Refresh Interval	60 (default)
Async Delay	15 (default)
Last pSnapId	0
Display Home Snapshots	no
Number of Gateway Flush Threads	4
Prefetch Threshold	0 (default)

Eviction Enabled	yes (default)
IO Flags	263168 (fastCreate)

Note: RPO or peer snapshot feature is not supported when the **afmFastCreate** parameter is set on an AFM or AFM-DR fileset.

```
# mmpsnap fs create -j fileset
```

The following error is displayed:

```
mmpsnap: [E] Peer snapshot creation failed. Error code 22.  
mmpsnap: Command failed. Examine previous error messages to determine cause.
```

Support of linking GPFS dependent fileset

Active File Management (AFM) supports to create and link a dependent fileset inside an independent fileset.

AFM can be enabled only on a GPFS independent fileset. However, the user can create a GPFS dependent fileset by using inode space of an AFM or AFM-DR fileset and link this dependent fileset under the AFM or AFM-DR fileset for the data synchronization. All data and metadata operations that are generated inside the dependent fileset, which is linked inside AFM fileset, work seamlessly. The user can create data inside the AFM or AFM-DR fileset path and the linked dependent fileset path. AFM synchronizes the created data to the home or secondary fileset under the same path. AFM queues operations that come under the root path of the AFM fileset and the linked path of the dependent fileset and flushes the operations to the home. Also, during the prefetch, AFM can bring data from home dependent fileset path to the AFM fileset that has the dependent fileset that is linked at same path. The path location must be the same at the cache or primary and the home or secondary site.

All I/O requests that are generated under the linked dependent fileset use the same gateway node that is assigned to the AFM or AFM-DR fileset. The gateway node maintains the directory structure during the data synchronization and keeps the home or secondary in sync.

To define an AFM relationship between a cache or primary and a home or secondary, stop the specific AFM or AFM-DR fileset, and then link the dependent fileset at the same junction path under the AFM or AFM-DR fileset. That is, the relative path under the AFM or AFM-DR fileset at both sides must be the same so that all operations generated at one site sync to the appropriate path at the other site. When the dependent fileset is linked inside the AFM or AFM-DR fileset, the user must issue **Start** command to start the replication.

Prerequisite for linking dependent fileset under an AFM or AFM-DR fileset:

1. Set up a dependent fileset at both cache or primary and home or secondary sites and then start replication. Because AFM does not replicate dependent fileset information between cache or primary and home or secondary sites.
2. Ensure that the dependent fileset link path (junction path) is the same relative path to the AFM or AFM-DR fileset at both sites.

That is, the linking of dependent fileset inside AFM or AFM-DR fileset must be identical at both sites.

3. Create and link dependent fileset at a new location to perform failover or **changeSecondary** operation on the AFM or AFM-DR fileset, which has the linked dependent fileset.

Note: Ensure that the failover or change secondary site has an identical path to the cache or primary for the data synchronization before you perform the failover or **changeSecondary** operation.

4. Run the **mmafmconfig enable** command on each dependent fileset that is linked inside an independent fileset at the home/secondary.

To link a dependent fileset inside an AFM or AFM-DR fileset, do the following steps:

1. Create an AFM or AFM-DR fileset and set up an AFM relationship.
2. Create a dependent fileset by using inode space of the created AFM or AFM-DR fileset under which you want to link the dependent fileset at both sites.

3. Stop the specific AFM or AFM-DR fileset at the cache or primary.
4. Link the dependent fileset inside the AFM or AFM-DR fileset at the same relative path at both sites so that the directory structure is the same across the sites.
5. Start the AFM or AFM-DR fileset.

Note: To unlink the dependent fileset, an AFM or AFM-DR fileset must be stopped first.

Example

1. Create an AFM or AFM-DR fileset and set up an AFM relationship.

```
# mmcrfileset <fs> <AFM/AFM-DR fileset> -p afmMode=<mode>,afmTarget=<AFMTarget> --inode-space New
```

```
# mmlinkfileset <fs> <AFM/AFM-DR fileset> -J /< GPFS_PATH >/<AFM|AFM-DR link path>
```

For more information, see the topic *Administering AFM* in the *IBM Spectrum Scale: Administration Guide*.

2. Create a dependent fileset by using inode space of the created AFM or AFM-DR fileset of the created AFM or AFM-DR fileset.

```
# mmcrfileset <fs> <dep-fileset> --inode-space <AFM/AFM-DR Fileset>
```

3. Stop the AFM or AFM-DR fileset at the cache.

```
# mmafmctl <fs> stop -j <AFM/ADR Fileset>
```

For more information, see [“Stop and start replication on a fileset” on page 90](#).

4. Link the dependent fileset at the relative junction path at both sites.

```
# mmlinkfileset <fs> <dep-fileset> -J <GPFS_PATH/AFM|AFM-DR Path/Dependent Fileset>
```

5. Start the AFM or AFM-DR fileset at the cache or primary.

```
# mmafmctl <fs> start -j <AFM/ADR Fileset>
```

For more information, see [“Stop and start replication on a fileset” on page 90](#).

Note: This feature supports linking only the GPFS dependent fileset that is under the AFM or AFM-DR fileset root path. This feature does not allow linking the nested levels of the dependent fileset under the AFM or AFM-DR fileset.

Force flushing contents before Async Delay

Requests in the queue of a writable cache (SW/IW) can be flushed to the home before they get flushed automatically after Async Delay by using the following **mmafmctl flushPending** command.

```
mmafmctl Device flushPending [-j FilesetName [--list-file ListFile]] [-s LocalWorkDirectory]
```

For more information, see the **mmafmctl** command in *IBM Spectrum Scale: Command and Programming Reference*.

Parallel data transfers

Parallel data transfer improves the AFM data transfer performance.

To help the primary gateway exchange large files with the home cluster, a cache cluster can be configured to leverage all the gateways defined in the cluster. When using NFS for AFM data transfers multiple NFS servers are required at the home cluster. All NFS servers on the home cluster must export the home path using the same parameters.

In a cache cluster, using NFS for AFM data transfer, each gateway node can be mapped to a specific NFS server at home. A map replaces the NFS server name in the **AFMTarget** parameter. Creating an export server map can be used to define more than one NFS server and map those NFS servers to specific AFM gateways. A map can be changed without modifying the **afmTarget** parameter for a fileset, and needs fileset relink or file system remount for the map change to take effect. Use the *mmafmconfig* command in *IBM Spectrum Scale: Command and Programming Reference* to define, display, delete, and update mappings.

To define multiple NFS servers for an **AFMTarget** parameter and use parallel data transfers:

1. Define a mapping.
2. Use the mapping as the **AFMTarget** parameter for one or more filesets.
3. Update parallel read and write thresholds, in chunk size, as required.

The following example shows a mapping for NFS target, assuming four cache gateway nodes hs22n18, hs22n19, hs22n20, and hs22n21, mapped to two home NFS servers js22n01 and js22n02 (192.168.200.11 and 192.168.200.12) and then creating SW filesets by using this mapping.

Define the mapping:

```
# mmafmconfig add mapping1 --export-map js22n01/hs22n18,js22n02/hs22n19
```

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

The syntax followed here is -

```
mmafmconfig {add | update} MapName --export-map ExportServerMap
```

```
# mmafmconfig add mapping2 --export-map js22n02/hs22n20,js22n01/hs22n21
```

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

```
# mmafmconfig show
```

```
Map name:          mapping1
Export server map:  192.168.200.12/hs22n19.gpfs.net,192.168.200.11/hs22n18.gpfs.net

Map name:          mapping2
Export server map:  192.168.200.11/hs22n20.gpfs.net,192.168.200.12/hs22n21.gpfs.net
```

#Create filesets by using these mappings:

```
mmcrfileset gpfs1 sw1 --inode-space new -p afmMode=sw,afmTarget=nfs://mapping1/gpfs/gpfs2/swhome
mmcrfileset gpfs1 ro1 --inode-space new -p afmMode=ro,afmTarget=nfs://mapping2/gpfs/gpfs2/swhome
```

The syntax followed here is -

```
mmcrfileset <FS> <fset_name> -p afmMode=<AFM Mode>,
afmTarget=<protocol>://<Mapping>/<remoteFS_Path>/<Target> --inode-space new
```

All gateway nodes other than the primary gateway that is defined in a mapping are called participating gateway nodes. The primary gateway of a cache fileset communicates with each of the participating gateway nodes, depending on their availability. When parallel data transfer is configured, a single data transfer request is split into multiple chunks. Chunks are sent across to the participating gateway nodes in parallel for transfer to, or from home by using the respective NFS servers. Primary gateway processes the replies from all the participating gateway nodes, handles all data transfer failures, and coordinates activities until all data transfer is completed. If any participating gateway node fails, the primary gateway

attempts to retry the failed task on the next available gateway and generates an error message in the IBM Spectrum Scale log.

Note: The parallel data transfer does not work in failover cases. This parallel data transfer works when the fileset state is moved to the dirty state.

Parallel reads and writes are effective on files with sizes larger than those specified by the parallel threshold. The threshold is defined by using **afmParallelWriteThreshold** and **afmParallelReadThreshold** parameters, and is true for all types of files except reads on sparse files and files with partial file caching enabled, which is served only by the Primary gateway without splitting.

Use the **afmParallelWriteChunkSize** and **afmParallelReadChunkSize** parameters to configure the size of each chunk.

Some more functions are as under -

1. While using native NSD protocol; if a fileset is created without any mapping, all gateway nodes are used for parallel data transfer.
2. While using NFS protocol, if more than one gateway node is mapped to the same NFS server, only one performs a read task. However, a write task is split among all the gateway nodes.
3. One gateway node cannot be mapped to more than one NFS server.
4. Changes in the active mapping take effect after fileset re-link or file system remount.
5. If mapping is not specified or if mapping does not match, data cannot be transferred by using parallel data transfers and normal data transfer function is used.
6. Gateway designation can be removed from a node only if that node is not defined in any mapping.

Note: If an AFM home is a mix of architectures (x86 and ppc), parallel data transfer works only for the set of nodes that belong to any one architecture, depending on which architecture serves the data transfer first.

This feature can be combined with the “Parallel data transfer using multiple remote mounts” on page 61 feature to obtain better data transfer performance within an AFM cache and an AFM home. Both features use the same AFM gateway node mapping that is defined by using the **mmafmconfig** command. These features are independent of each other and you can set these features by considering what suits better for a workload.

Parallel data transfer using multiple remote mounts

Parallel data transfer by using the multiple remote mounts on the gateway node improves the AFM data transfer performance by sending unique data operations across multiple NFS server mounts.

A cache cluster can be configured for parallel data transfers by using multiple remote mounts when a target site has two or more NFS servers. This feature uses a mapping between the gateway nodes at a cache (primary) site and the multiple NFS serving nodes at a home (secondary) cluster by exporting the target path. When you use NFS as a backed protocol for AFM data transfers, multiple NFS servers can be used at the home cluster. All NFS servers at the home cluster must export the same target path by using the same parameters for a specific fileset.

The AFM parallel mounts feature uses the same mapping infrastructure as the parallel data transfer. The parallel data transfer feature is used to transfer files bigger than the read or write threshold values. However, when the AFM parallel mounts feature is enabled, NFS server nodes can send and receive data in parallel by using the available bandwidth.

In a cache cluster, by using NFS for AFM data transfer, each gateway node can be mapped to a specific NFS server at a home. Creating an export server map can be used to define more than one NFS server and map those NFS servers to specific AFM gateways.

The AFM parallel mounts feature can be enabled or disabled by setting the **afmParallelMounts** parameter in the **mmchconfig** command or the **mmchfileset** command.

This parameter is available at the global cluster level for the **mmchconfig** command and at the fileset level for the **mmchfileset** command.

Do the following steps to configure and set up AFM parallel data transfer by using multiple remote mounts:

1. Enable the **afmParallelMounts** parameter at the cluster level.

Set up this tunable at the cluster level to enable the **afmParallelMounts** parameter for all filesets by using the following command:

```
# mmchconfig afmParallelMounts=yes -i
```

2. Define a mapping by identifying gateway nodes and NFS servers.

To create a mapping, the user must identify the gateway node at the cache cluster and NFS servers at the home site. This mapping enables mounting remote export from two or more NFS servers at the gateway node and synchronizes unique data by using multiple remote mounts. The user can create an AFM or ADR fileset by using this mapping as the **AFMTarget** parameter.

3. Start data synchronization between the home and cache clusters, this synchronization uses multiple remote mounts at gateway nodes.

User can also enable this parameter for a specific AFM fileset. The user can set the **afmParallelMounts** parameter by issuing the following command:

```
# mmchfileset <FS> -j <AFM/ADR fileset> -p afmParallelMounts=yes
```

Example

The following example shows a mapping for an NFS target. Two cache gateway nodes *cachegw_n1* and *cachegw_n2* are mapped to two home NFS servers *homenfs_n1* and *homenfs_n2* (192.168.200.11 and 192.168.200.12), and then SW filesets are created by using this mapping for the parallel data transfer by using multiple remote mounts.

1. Create a mapping between the NSF export server and the NFS server.

```
# mmafmconfig add mapping1 --export-map homenfs_n1/cachegw_n1,homenfs_n2/cachegw_n2
mmafmconfig: Command successfully completed
mmafmconfig: Propagating the cluster configuration data to all affected nodes. This is an
asynchronous process.
```

The syntax is as follows:

```
# mmafmconfig {add | update} MapName --export-map ExportServerMap
```

```
# mmafmconfig show
Map name:          mapping1
Export server map: 192.168.200.12/cachegw_n2.gpfs.net,192.168.200.11/cachegw_n1.gpfs.net
```

2. Create filesets by using *mapping1* and link the fileset.

```
# mmcrfileset gpfs1 sw1 --inode-space new -p afmMode=sw,afmTarget=nfs://mapping1/gpfs/gpfs2/
swhome
Fileset sw1 created with id 14 root inode 3670019.
```

3. Issue the following command to set parameter for a specific fileset:

```
#mmchfileset gpfs1 sw1 -p afmParallelMounts=yes
Fileset sw1 changed.
```

```
# mmlinkfileset fs1 sw1 -J /gpfs1/sw1
Fileset sw1 linked at /gpfs1/sw1
```

By using parallel remote mounts, all the data and metadata operations are uniquely replicated to the target site by queuing them to the unique remote mounts or channels. This method keeps the queue or operation processing completely within the primary gateway node for the fileset and involves multiple NFS servers that are defined in the mapping from the remote cluster. Primary gateway tracks operation

that is played to different NFS mounts. This setup supports both write from the cache and read from the remote cluster.

This feature can be combined with the “Parallel data transfers” on page 59 feature to obtain better data transfer performance within an AFM cache and an AFM home. Both features use the same AFM gateway node mapping that is defined by using the **mmafmconfig** command. These features are independent of each other and you can set these features by considering what is suited best for a workload.

Partial file caching

With partial file caching, the cache can fetch only the blocks that are read and not the entire file, thereby using network and local disk space more efficiently. This caching is useful when an application does not need to read the whole file. Partial file caching is enabled on an IBM Spectrum Scale block boundary.

Partial file caching is controlled by the **afmPrefetchThreshold** parameter that can be updated by using the **mmchfileset** command. The default value of this parameter is 0. Complete file caching and all blocks of a file are fetched after any three blocks are read by the cache and the file is marked as cached because of this value. This value is useful for sequentially accessed files that are read in their entirety, such as image files, home directories, and development environments.

The valid **afmPrefetchThreshold** values are in the range 1 – 100. This parameter value specifies the file size percentage that must be cached before the rest of the data blocks are automatically fetched into the cache. A large value is suitable for a file that is accessed partially.

An **afmPrefetchThreshold** value of 100 disables full file prefetching. This value caches only the data blocks that are read by the application. This value is useful for large random-access files that are either too large to fit in the cache or are never expected to be read in their entirety. When all data blocks are available in the cache, the file is marked as cached.

For sparse files, the percentage for prefetching is calculated as the ratio of the size of data blocks that is allocated in the cache and the total size of data blocks on the home. Holes in the home file are not considered in the calculation.

Writes on partially cached files

If a write is queued on a file that is partially cached, then the complete file is cached first. Only then the write is queued on the file. Appending to a partially cached file does not cache the whole file. In the LU mode alone, the write inset or append on a file that is cached partially caches the whole file even if the prefetch threshold is set on the fileset.

Note: As partial file caching is not compatible with earlier versions, all nodes must be on GPFS 3.5.0.11 or later.

Prefetch

Prefetch fetches the file metadata (inode information) and data from home before an application requests the contents.

Prefetch is a feature that allows fetching the contents of a file into the cache before actual reads.

Prefetching files before an application starts can reduce the network delay when an application requests a file. Prefetch can be used to proactively manage WAN traffic patterns by moving files over the WAN during a period of low WAN usage.

Prefetch can be used to do the following tasks:

- Populate metadata
- Populate data
- View prefetch statistics

Use the following command to perform these activities:

```
mmafmctl Device prefetch -j FilesetName [-s LocalWorkDirectory]
[--retry-failed-file-list|--enable-failed-file-list]
[{--directory LocalDirectoryPath | --dir-list-file DirListfile [--policy]} [--nosubdirs]]
[{--list-file ListFile | --home-list-file HomeListFile} [--policy]]
```

```
[--home-inode-file PolicyListFile]
[--home-fs-path HomeFilesystemPath]
[--metadata-only] [--gateway Node]
[--readdir-only] [--force] [--prefetch-threads nThreads]
```

For more information about the command, see *mmafmctl* command in *IBM Spectrum Scale: Command and Programming Reference*. If no options are given for prefetch, the statistics of the last prefetch command that is run on the fileset are displayed.

--metadata-only

Prefetches only the metadata and not the actual data. This option is useful in migration scenarios. This option requires the list of files whose metadata is to be populated. It must be combined with a list file option.

--list-file ListFile

The specified file contains a list of files that need to be pre-populated, one file per line. All files must have fully qualified path names. If the list of files to be prefetched have file names with special characters, then a policy must be used to generate the listfile. This list file must be edited manually to remove all other entries except the file names. The list of files can be:

1. Files with fully qualified names from cache
2. Files with fully qualified names from home
3. List of files from the home that are generated by using the policy. The file must not be edited.

--enable-failed-file-list

Turns on generating a list of files that failed during prefetch operation at the gateway node. The list of files is saved as `.afm/.prefetchedfailed.list` under the fileset. Failures that occur during processing are not logged in `.afm/.prefetchedfailed.list`. If you observe any errors during processing (before queuing), you might need to correct the errors and rerun prefetch.

--policy

Specifies that the list-file or home-list-file is generated by using a GPFS Policy by which sequences like `'\'` or `'\n'` are escaped as `'\\'` and `'\\n'`. If this option is specified, input file list is treated as already escaped. The sequences are unescaped first before queuing for prefetch operation.

Note: This option can be used only if you are specifying list-file or home-list-file.

--directory LocalDirectoryPath

Specifies path to the local directory from which you want to prefetch files. A list of all files in this directory and all its subdirectories is generated, and queued for prefetch. You can either specify `--directory` or `--dir-list-file` with **mmafmctl prefetch**. The `--policy` option can be used only with `--dir-list-file` and not with `--directory`.

For example,

```
# mmafmctl fs1 prefetch -j fileset1 --dir-list-file /tmp/file1 --policy
```

The following example includes methods to name a directory for the `--directory` option, when the directory name contains special characters:

- When a directory name does not have terminal escape sequences, keep the absolute directory path within double quotation marks (" ").

```
# mmafmctl fs2 prefetch -j roTestPrefetch_GPFS --directory "/gpfs/fs2/roTestPrefetch1/Dir_a\!h@j#k%l^k&78*9'\''"
```

A sample output is as follows:

```
mmafmctl: Performing prefetching of fileset: roTestPrefetch_GPFS
mmafmctl(2020-04-13 02:35:39): Listing all files of directory "/gpfs/fs2/roTestPrefetch1/Dir_a\!h@j#k%l^k&78*9'\''"
      Queued      Failed      TotalData
                        (approx in Bytes)
          25          0      131072000
prefetch successfully queued at the gateway.
```



```
(2020-04-13 02:35:41): Listed all files of directory "/gpfs/fs2/roTestPrefetch1/Dir_a\!h@j#k%l^k&78*9'\''"
```

- When the directory name has terminal escape sequences, do not keep the directory path within double quotation marks. The terminal auto-fills the escape sequences in the directory name when you press the <Tab> two times.

```
# mmafmctl fs2 prefetch -j roTestPrefetch_GPFS --directory /gpfs/fs2/roTestPrefetch1/Dir_a\\!h@j#k%l^k&78*9'\''/'
```

A sample output is as follows:

```
mmafmctl: Performing prefetching of fileset: roTestPrefetch_GPFS
mmafmctl(2020-04-13 02:39:58): Listing all files of directory "/gpfs/fs2/roTestPrefetch1/Dir_a\!h@j#k%l^k&78*9'\''/"
      Queued      Failed      TotalData
                        (approx in Bytes)
      25          0          131072000
prefetch successfully queued at the gateway.
mmafmctl(2020-04-13 02:40:00): Listed all files of directory "/gpfs/fs2/roTestPrefetch1/Dir_a\!h@j#k%l^k&78*9'\''/"
```

- When press <Tab> two times to include the escape sequences a directory name and keep the directory path within double quotation marks, the prefetch operation fails. The prefetch operation fails because the unescape of terminal escaped characters in the directory name is not performed.

```
# mmafmctl fs2 prefetch -j roTestPrefetch_GPFS --directory "/gpfs/fs2/roTestPrefetch1/Dir_a\\!h@j#k%l^k&78*9'\''/'
```

A sample output is as follows:

```
mmafmctl: Performing prefetching of fileset: roTestPrefetch_GPFS
runPrepopSubcommand: Unexpected error from missing or incorrect prepop input path.
Return code: 1
mmafmctl: Command failed. Examine previous error messages to determine cause.
```

--dir-list-file *DirListFile*

This parameter enables prefetching individual directories under AFM fileset. Input file specifies the unique path to a directory that you want to prefetch. AFM generates a list of files under the specified directory and subdirectories and queues it to the gateway Node. The input file can also be a policy-generated file for which you need to specify --policy

--nosubdirs

This option restricts the recursive behavior of --directory and --dir-list-file and prefetches only until the specified level of directory. If you specify this parameter, subdirectories under the directory are not prefetched. This parameter is optional and can be used only with --directory and --dir-list-file.

For example,

```
# mmafmctl fs1 prefetch -j fileset1 --directory /gpfs/fs1/fileset1/dir1 --nosubdirs
```

```
# mmafmctl fs1 prefetch -j fileset1 --dir-list-file /tmp/file1 --policy --nosubdirs
```

--retry-failed-file-list

Allows retrying prefetch of files that failed in the last prefetch operation. The list of files to retry is obtained from .afm/.prefetchedfailed.list under the fileset.

Note: To use this option, you must enable generating a list of failed files. Add **--enable-failed-file-list** to the command first.

--home-list-file *HomeListFile*

The specified file contains a list of files from home that need to be pre-populated, one file per line. All files must have fully qualified path names. If the list of files to be prefetched have file names with special characters, then a policy must be used to generate the listfile. A policy-generated file

must be edited manually to remove all other entries except the file names. As of version 4.2.1, this option is deprecated. The **--list-file** option removes all other entries except the file names.

--home-inode-file *PolicyListFile*

The specified file contains the list of files from home that need to be pre-populated in the cache and this file is generated by using policy. This file must not be edited manually. This option is deprecated. The **-list-file** option removes all other entries except the file names.

--home-fs-path *HomeFileSystemPath*

Specifies the full path to the fileset at the home cluster and can be used along with **-list-file**. You must use this option, when in the NSD protocol the mount point on the gateway nodes of the **afmTarget** filesets does not match the mount point on the Home cluster. For example, the home file system is mounted on the home cluster at `/gpfs/homefs1`. The home file system is mounted on the cache by using NSD protocol at `/gpfs/remotefs1`.

For example,

```
# mmafmctl gpfs1 prefetch -j cache1 -list-file /tmp/list.allfiles --home-fs-path /gpfs/remotefs1
```

--readdir-only

Enables `readdir` operation on a dirty directory at the cache one last time and brings latest directory entries.

This option helps prefetching modified directory entries from the home, although the directory at the cache fileset was modified by the applications and AFM marked the dirty flag on the cache directory. This option overrides the dirty flag that is set when the data is modified at the local LU cache. In the LU mode, the dirty flag does not allow the `readdir` operation at the home and refreshes the directory file entries from the home.

This option helps in the migration process where new files were created at the home after the application was moved to the cache. The application already modified the directory and refresh intervals were disabled. AFM queues `readdir` one last time on the cache directory and brings entries of the created files to the cache.

The **afmReadDirOnce** parameter must be set on an AFM fileset, and directory and files refresh intervals must be disabled.

For example,

1. To set **afmRefreshOnce** on an AFM fileset, issue the following command:

```
# mmchfileset fs fileset -p afmRefreshOnce=yes
```

2. To check whether the **afmRefreshOnce** parameter value is set on an AFM fileset, issue the following command:

```
# mmlsfileset fs fileset -L --afm
```

A sample output is as follows:

```
Filesets in file system '<fs>':
Attributes for fileset <fileset>:
=====
Status                               Linked
Path                                GPFS_PATH/fileset
Id                                  37
Root inode                          3145731
Parent Id                           0
Created                             Wed Mar  4 12:23:47 2020
Comment
Inode space                          6
Maximum number of inodes             100352
Allocated inodes                     100352
Permission change flag               chmodAndSetacl
afm-associated                       Yes
Target                              nfs://home/fileset
Mode                                local-updates
```

File Lookup Refresh Interval	30 (default)
File Open Refresh Interval	30 (default)
Dir Lookup Refresh Interval	60 (default)
Dir Open Refresh Interval	60 (default)
Expiration Timeout	disable (default)
Last pSnapId	0
Display Home Snapshots	yes (default)
Number of Gateway Flush Threads	4
Prefetch Threshold	0 (default)
Eviction Enabled	yes (default)
I/O Flags	9216 (refreshOnce)

3. To run the prefetch operation for the readdir operation one last time, issue following command:

```
# mmadmctl fs prefetch -j fileset --directory /fileset_path/directory --readdir-only
```

--force

Enables forcefully fetching data from the home during the migration process. This option overrides any set restrictions and helps to fetch the data forcefully to the cache. This option must be used only to forcefully fetch the data that was created after the migration process completion.

For example,

```
# mmadmctl fs prefetch -j fileset --list-file listfile_path --force
```

--gateway Node

Allows selecting the gateway node that can be used to run the prefetch operation on a fileset, which is idle or less-utilized. This option helps to distribute the prefetch work on different gateway nodes and overrides the default gateway node, which is assigned to the fileset. It also helps to run different prefetch operations on different gateway nodes, which might belong to the same fileset or a different fileset.

For example,

```
# mmadmctl fs prefetch -j fileset --list-file listfile_path --gateway Node2
```

--prefetch-threads nThreads

Specifies the number of threads to be used for the prefetch operation. Valid values are 1 - 255. Default value is 4.

For example,

```
# mmadmctl fs prefetch -j fileset --list-file listfile_path --prefetch-threads 6
```

Prefetch is an asynchronous process and the fileset can be used while prefetch is in progress. Prefetch completion can be monitored by using the **afmPrepopEnd** callback event or looking at **mmadmctl Device prefetch** command with no options.

Prefetch pulls the complete file contents from home (unless the **--metadata-only** flag is used), so the file is designated as cached when it is prefetched. Prefetch of partially cached files caches the complete file.

Prefetch can be run in parallel on multiple filesets, although only one prefetch job can run on a fileset.

While a file is getting prefetched, it is not evicted.

If parallel data transfer is configured, all gateways participate in the prefetch process.

If the file system unmounts during prefetch on the gateway, issue the prefetch again.

Prefetch can be triggered on inactive filesets.

Directories are also prefetched to the cache if specified in the prefetch file. If you specify a directory in the prefetch file and if that directory is empty, the empty directory is prefetched to cache. If the directory contains files or subdirectories, you must specify the names of the files or subdirectories that you want to prefetch. If you do not specify names of individual files or subdirectories inside a directory, that directory is prefetched without its contents.

If you run the prefetch command with data or metadata options, statistics like queued files, total files, failed files, total data (in bytes) is displayed.

For example,

```
# mmafmctl FileSystem prefetch -j fileset --enable-failed-file-list --list-file /tmp/file-list
```

A sample output is as follows:

```
mmafmctl: Performing prefetching of fileset: <fileset>
Queued (Total) Failed TotalData (approx in Bytes)
0 (56324) 0 0
5 (56324) 2 1353559
56322 (56324) 2 14119335
```

Prefetch Recovery:

Note: This feature is disabled from IBM Spectrum Scale 5.0.2. If your cluster is running on an earlier version, prefetch recovery is possible.

If the primary gateway of a cache is changed while prefetch is running, prefetch is stopped. The next access to the fileset automatically retriggers the interrupted prefetch on the new primary gateway. The list file used when prefetch was initiated must exist in a path that is accessible to all gateway nodes. Prefetch recovery on a single-writer fileset is triggered by a read on some file in the fileset. Prefetch recovery on a read-only, independent-writer, and local-update fileset is triggered by a lookup or readdir on the fileset. Prefetch recovery occurs on the new primary gateway and continues where it left off. It looks at which files did not complete prefetch and it rebuilds the prefetch queue. Examples of messages in the mmfs.log are as follows:

```
Wed Oct 1 13:59:22.780 2014: [I] AFM: Prefetch recovery started for the file system gpfs1
fileset iw1.
mmafmctl: Performing prefetching of fileset: iw1
Wed Oct 1 13:59:23 EDT 2014: mmafmctl: [I] Performing prefetching of fileset: iw1
Wed Oct 1 14:00:59.986 2014: [I] AFM: Starting 'queue' operation for fileset 'iw1' in
filesystem '/dev/gpfs1'.
Wed Oct 1 14:00:59.987 2014: [I] Command: tspace /dev/gpfs1 1 iw1 0 257 42949 67295 0 0
1393371
Wed Oct 1 14:01:17.912 2014: [I] Command: successful tspace /dev/gpfs1 1 iw1 0 257
4294967295 0 0 1393371
Wed Oct 1 14:01:17.946 2014: [I] AFM: Prefetch recovery completed for the filesystem gpfs1
fileset iw1. error 0
```

1. Metadata population by using prefetch:

```
# mmafmctl fs1 getstate -j ro
```

A sample output is as follows:

Fileset Name	Fileset Target	Cache State	Gateway Node	Queue Length	Queue numExec
ro	nfs://c26c3apv1/gpfs/homefs1/dir3	Active	c26c4apv1	0	7

List Policy:
 RULE EXTERNAL LIST 'List' RULE 'List' LIST 'List' WHERE PATH_NAME LIKE '%'
 Run the policy at home:mmapplypolicy /gpfs/homefs1/dir3 -P px -f px.res -L 1 -N mount -I defer
 Policy creates a file which should be manually edited to retain only the file names. Thereafter
 this file is used at the cache to populate metadata.

```
# mmafmctl fs1 prefetch -j ro --metadata-only --list-file=px.res.list.List
```

A sample output is as follows:

```
mmafmctl: Performing prefetching of fileset: ro
Queued (Total) Failed TotalData
0 (2) 0 0
100 (116) 5 1368093971
116 (116) 5 1368093971

prefetch successfully queued at the gateway
```

Prefetch end can be monitored by using this event:
 Thu May 21 06:49:34.748 2015: [I] Calling User Exit Script prepop: event afmPrepopEnd,
 Async command prepop.sh.

The statistics of the last prefetch command can be viewed by running the following command:

```
mmafmctl fs1 prefetch -j ro
```

Fileset Name	Async Read (Pending)	Async Read (Failed)	Async Read (Already Cached)	Async Read (Total)	Async Read (Data in Bytes)
ro	0	1	0	7	0

2. Prefetch of data by giving list of files from home:

```
# cat /listfile1
```

A sample output is as follows:

```
/gpfs/homefs1/dir3/file1
/gpfs/homefs1/dir3/dir1/file1
```

```
# mmafmctl fs1 prefetch -j ro --list-file=/listfile1
```

A sample output is as follows:

mmafmctl: Performing prefetching of fileset: ro

Queued	(Total)	Failed	TotalData
			(approx in Bytes)
0	(2)	0	0
2	(2)	0	1368093971

```
# mmafmctl fs1 prefetch -j ro
```

A sample output is as follows:

Fileset Name	Async Read (Pending)	Async Read (Failed)	Async Read (Already Cached)	Async Read (Total)	Async Read (Data in Bytes)
ro	0	0	0	2	122880

3. Prefetch of data by using a list file, which is generated by using policy at home:

Inode file is created by using the policy at home, and must be used without editing manually.

```
List Policy:
RULE EXTERNAL LIST 'List' RULE 'List' LIST 'List' WHERE PATH_NAME LIKE '%'
```

For files with special characters, path names must be encoded with ESCAPE %.

```
RULE EXTERNAL LIST 'List' ESCAPE '%' RULE 'List' LIST 'List' WHERE PATH_NAME LIKE '%'
```

Run the policy at home:

```
# mmapplypolicy /gpfs/homefs1/dir3 -P px -f px.res -L 1 -N mount -I defer
```

```
# cat /lfile2
```

A sample output is as follows:

```
113289 65538 0 -- /gpfs/homefs1/dir3/file2
113292 65538 0 -- /gpfs/homefs1/dir3/dir1/file2
```

```
#mmafmctl fs1 prefetch -j ro --list-file=/lfile2
```

A sample output is as follows:

```
mmafmctl: Performing prefetching of fileset: ro
# mmafmctl fs1 prefetch -j ro -list-file=/lfile2

mmafmctl: Performing prefetching of fileset: ro

Queued (Total) Failed TotalData
      (2)      0      (approx in Bytes)
0          0
2          0      1368093971
```

4. Prefetch by using `--home-fs-path` option for a target with the NSD protocol:

```
# mmafmctl fs1 getstate -j ro2
```

A sample output is as follows:

Fileset Name	Fileset Target	Cache State	Gateway Node	Queue Length	Queue numExec
ro2	gpfs:///gpfs/remotefs1/dir3	Active	c26c4apv1	0	7

```
# cat /lfile2
```

A sample output is as follows:

```
113289 65538 0 -- /gpfs/homefs1/dir3/file2
113292 65538 0 -- /gpfs/homefs1/dir3/dir1/file2
```

```
# mmafmctl fs1 prefetch -j ro2 -list-file=/lfile2 --home-fs-path=/gpfs/homefs1/dir3
```

A sample output is as follows:

```
mmafmctl: Performing prefetching of fileset: ro2

Queued (Total) Failed TotalData
      (2)      0      (approx in Bytes)
0          0
2          0      113292
```

```
# mmafmctl fs1 prefetch -j ro2
```

A sample output is as follows:

Fileset Name	Async Read (Pending)	Async Read (Failed)	Async Read (Already Cached)	Async Read (Total)	Async Read (Data in Bytes)
ro2	0	0	0	2	122880

Peer snapshot -psnap

The peer snapshot function provides a snapshot at home and cache separately, ensuring application consistency on both sides (cache and home).

When you take a peer snapshot, it creates a snapshot in the cache fileset then queues a snapshot to be executed at the home. When all of the queued requests outstanding at the time the snap was performed in cache have been flushed to the home fileset so that the home data is consistent with cache, a snapshot of the corresponding home fileset is performed. The result is a pair of peer snapshots, one at the cache and one at the home. Both refer to the same copy.

Peer snapshots are created using the **mmpsnap** command on the cache. To create a peer snapshot, the cache fileset must be in the active state. The last successful snapshot is saved and can be viewed by running the **mm1ssnapshot** command at home or cache. Multiple outstanding peer snapshots can be queued on the gateway. Use the **mmpsnap** command to ensure that both the cache and home snapshots are removed. The **mmpsnap** command works only if the home cluster has run the **mmafmconfig enable ExportPath** command. The **mmpsnap** command can be used only in an SW cache. It cannot be used for RO, IW, or LU caches.

Note: Do not use the **mmdeletesnapshot** command to delete peer snapshots.

If the cache fileset is disconnected from the home fileset when the cache snapshot is created, the cache records that the peer snapshot on the home fileset has not been created. When connection is restored, it attempts to create the snapshot.

Peer snapshots are not allowed on a SW cache that uses the NSD protocol for communicating with home.

1. To create a fileset level snapshot in cache of a single-writer fileset called **sw** in file system **fs1**, run the following command:

```
mmpsnap fs1 create -j sw
```

The system displays the following output:

```
Writing dirty data to disk.
Quiescing all file system operations.
Writing dirty data to disk again.
Snapshot psnap-13882361812785649740-C0A80E85:4F44B305-59-12-03-01-02-27-28 created with id 8.
Snapshot psnap-13882361812785649740-C0A80E85:4F44B305-59-12-03-01-02-27-28 created at the
satellite.
Core snapshot has been queued.
```

2. To view the snapshot, run the following command:

```
mmllsnapshot fs1 -j sw
```

The system displays the following output:

```
Snapshots in file system fs1:
Directory SnapId Status Created Fileset
psnap-13882361812785649740-C0A80E85:4F44B305-59-12-03-01-02-27-28 8 Valid Thu Oct 27
02:27:29 2016 sw
```

3. To view the snapshot at home, run the following command at home.:

```
mmllsnapshot fs1
```

The system displays the following output:

```
Snapshots in file system fs1:
Directory SnapId Status Created Fileset
psnap-13882361812785649740-C0A80E85:4F44B305-59-12-03-01-02-27-28 8 Valid Thu Oct 27
02:23:16 2016
```

Gateway node failure and recovery

When the primary gateway of a fileset fails, another gateway node takes over the ownership of the fileset.

Gateway node failures are not catastrophic and do not result in the loss of data or the loss of the ability of AFM to communicate with the home cluster with updates and revalidations.

AFM internally stores all the information necessary to replay the updates made in the cache at the home cluster. When a gateway node fails, the in-memory queue is lost. The node rebuilt the queue in memory by taking over for the failed gateway. The process of rebuilding the queue is called recovery. As an administrator, ensure that you create sufficient disk space in `/var/mmfs/afm` for a smooth recovery. For the recovery of 1 million files or inodes, approximately 250 MB disk space is required in `/var/mmfs/afm`.

During recovery, outstanding cache updates are placed on the in-memory queue and the gateway starts processing the queue. AFM collects the pending operations by running a policy scan in the fileset. AFM uses the policy infrastructure in IBM Spectrum Scale to engage all the nodes that are mounting the file system. Pending requests are queued in a special queue that is called the priority queue. The priority queue is different from the normal queue where normal requests get queued. After the priority queue is flushed to home, the cache and home become synchronized and recovery is said to be completed and the cache returns to an Active state. In some cases, the files or directories that are deleted at cache might not be deleted at home. Therefore, the files or directories that are not deleted remain at the home. However, the recovery operations remain unaffected.

The beginning and end of the AFM recovery process can be monitored by using the **afmRecoveryStart** and **afmRecoveryEnd** callback events.

Recovery is used only for the single-writer mode filesets and the independent-writer mode filesets. It is triggered when the cache fileset attempts to move to the Active state, for example when the fileset is accessed for the first time after the failure.

Recovery can run in parallel on multiple filesets, although only one instance can run on a fileset at a time. The time taken to synchronize the contents of cache with home after the recovery of a gateway node depends on the number of files in the fileset and the number of outstanding changes since the last failure.

During the recovery or resync operation on an AFM fileset, complete data synchronization from the cache to the home might take some time. It depends on various factors such as the total amount of data, network, disk I/O. You can check the synchronization status by using the **--write-stats** and **--read-stats** options.

```
# mmafmctl fs1 getstate -j filesetSW1 --write-stats
```

A sample output is as follows:

Fileset Name	Total Written Data (Bytes)	N/w Throughput (KB/s)	Total Pending Data to Write(Bytes)	Estimated Completion time
-----	-----	-----	-----	-----
filesetSW1	98359590	68	22620600	5 (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.

In the multiple AFM caches environment, recovery is triggered after a failure. To limit the maximum number of AFM or AFM-DR filesets that can perform recovery at a time, set the **afmMaxParallelRecoveries** parameter. The recovery process is run on the number of filesets that you specify for **afmMaxParallelRecoveries**. The number of filesets that are specified in **afmMaxParallelRecoveries** are accessed for recovery. After recoveries are complete on these filesets, the next set of filesets is accessed for recovery. By default, **afmMaxParallelRecoveries** is set to 0, and the recovery process is run on all filesets. Specifying **afmMaxParallelRecoveries** restricts the number of recoveries, thus conserving hardware resources. For more information, see *Configuration parameters for AFM in IBM Spectrum Scale: Administration Guide*.

Peer snapshots that are created in cache and queued to home might get lost due to gateway node failure. These peer snapshots cannot be recovered through AFM recovery. For more information, see [“Peer snapshot -psnap” on page 70](#).

If any AFM-enabled fileset has no updates, the failure of a gateway node is harmless and application nodes do not experience delays. During recovery, application requests to all AFM filesets are momentarily blocked.

The following example indicates changes in the AFM fileset state during recovery:

```
node2:/gpfs/cache/fileset_SW # mmafmctl fs1 getstate
Fileset Name Fileset Target Cache State Gateway Node Queue Length Queue numExec
-----
fileset_SW nfs://node4/gpfs/fshome/fset001 FlushOnly node2 0 0

node2:/gpfs/cache/fileset_SW # mmafmctl fs1 getstate
Fileset Name Fileset Target Cache State Gateway Node Queue Length Queue numExec
-----
fileset_SW nfs://node4/gpfs/fshome/fset001 Recovery node2 0 0

node2:/gpfs/cache/fileset_SW # mmafmctl fs1 getstate
```


Fileset Name	Fileset Target	Cache State	Gateway Node	Queue Length	Queue numExec
fileset_SW	nfs://node4/gpfs/fshome/fset001	Active	node2	0	3

For more information, see the *mmafmctl* command in *IBM Spectrum Scale: Command and Programming Reference*.

An example of the messages in *mmfs.log* is as follows:

```
Thu Oct 27 15:28:15 CEST 2016: [N] AFM: Starting recovery for fileset 'fileset_SW' in fs 'fs1'
Thu Oct 27 15:28:15 CEST 2016: mmcommon afmrecovery invoked: device=fs1 filesetId=1....
Thu Oct 27 15:28:16 CEST 2016: [N] AFM: mmafmlocal /usr/lpp/mmfs/bin/mmapplypolicy...
Thu Oct 27 15:28:31.325 2016: [I] AFM: Detecting operations to be recovered...
Thu Oct 27 15:28:31.328 2016: [I] AFM: Found 2 update operations...
Thu Oct 27 15:28:31.331 2016: [I] AFM: Starting 'queue' operation for fileset 'fileset_SW' in
filesystem 'fs1'.
Thu Oct 27 15:28:31.332 2016: [I] Command: tpspcache fs1 1 fileset_SW 0 3 1346604503 38 0 43
Thu Oct 27 15:28:31.375 2016: [I] Command: successful tpspcache fs1 1 fileset_SW 0 3 1346604503
38 0 43
Thu Oct 27 15:28:31 CEST 2016: [I] AFM: Finished queuing recovery operations for /gpfs/cache/
fileset_SW
```

Failures during recovery

Filesets can be in recovery state and might not complete recovery due to some conditions. The fileset might go to **Dropped** or **NeedsResync** state. This fileset state implies that recovery is failed.

The *mmfs.log* might contain the following lines: AFM: File system fs1 fileset adrSanity-160216-120202-KNFS-TC11-DRP encountered an error synchronizing with the remote cluster. Cannot synchronize with the remote cluster until AFM recovery is executed. remote error 28.

After the recovery fails, the next recovery is triggered after 120 seconds or when some operation is performed on the SW or IW fileset. After a successful recovery, modifications at the cache are synchronized with the home and the fileset state is 'Active'.

The following checks are needed from the administrator to ensure that the next recovery is successful:

1. Check the inode or block quota on cache and at home.
2. Ensure that home is accessible. Remount home file system and restart NFS at home.
3. Ensure that memory is not reached. If memory is reached, increase **afmHardMemThreshold**.
4. Check network connectivity with home.
5. If recovery keeps failing as eviction is triggered due to exceeding block quotas, increase block quotas or disable eviction to get recovery working.

Cache eviction

Cache eviction is a feature where file data blocks in the cache are released when a fileset usage exceeds the fileset soft quota, and space is created for new files.

The process of releasing blocks is called eviction. However, file data is not evicted if the file data is dirty. A file whose outstanding changes are not flushed to home is a dirty file.

You can use automatic cache eviction or define your own policy to decide which file data needs to be evicted. To automatically enable cache eviction, define a fileset soft quota on the cache fileset. Eviction starts when fileset usage reaches the soft quota limit. A time lag might result when an eviction is triggered and the data is evicted. Tuning the soft and hard quota limits minimizes application delays because the data is being cached at a faster rate than it is being evicted.

Cache eviction based on inode quotas is not supported. Cached data from partially fetched files can be evicted manually by using **--file** parameter of **mmafmctl** command. When files are evicted from cache, all data blocks of those files are cleared. The files are then uncached. When a read operation is performed on the files the next time, the files fetch data from home.

Cache eviction is enabled by default on all AFM nodes and is controlled by the **afmEnableAutoEviction** parameter, and fileset block quota. Cache eviction can also be manually triggered by using the **mmafmctl evict** command. When a file is evicted, file data is removed from the cache, but the inode stays in the cache. Using eviction, you can build environments, where all objects from home are available but running in limited amount of space.

For example, a cache can be created in flash storage. File eviction opens a powerful method of providing small but high speed and low latency cache filesets to clusters.

Manual eviction can be done by using the **mmafmctl evict** command as follows:

```
mmafmctl Device evict -j FilesetName
    [--safe-limit SafeLimit] [--order {LRU | SIZE}]
    [--log-file LogFile] [--filter Attribute=Value ...]
    [--list-file ListFile] [--file FilePath]
```

For more information, see *mmafmctl* command in *IBM Spectrum Scale: Command and Programming Reference*.

This option is applicable for RO/SW/IW/LU filesets. This command can be run manually or run in a script with a custom policy to implement a custom eviction policy. Options can be combined.

--safe-limit *SafeLimit* – This parameter is mandatory for the manual eviction, for order and filter attributes. It specifies target quota limit that is used as the low water mark for eviction in bytes – the value must be less than the soft limit. This parameter can be used alone or can be combined with one of the following parameters (order or filter attributes). Specify the parameter in bytes.

--order *LRU* | *SIZE* – The order in which files are to be chosen for eviction: LRU - Least recently used files are to be evicted first. SIZE - Larger-sized files are to be evicted first.

--log-file *Log File* – The file where the eviction log is to be stored. By default no logs are generated.

--filter *Attribute=Value* – The attributes that you can use to control the way data is evicted from the cache. Valid attributes are: FILENAME=*File Name* - The name of a file to be evicted from the cache. This option uses an SQL-type search query. If the same file name exists in more than one directory, the cache evicts all the files with that name. The complete path to the file must not be given here. MINFILESIZE=*Size* - The minimum size of a file to evict from the cache. This value is compared to the number of blocks that are allocated to a file (KB_ALLOCATED), which might differ slightly from the file size. MAXFILESIZE=*Size* - The maximum size of a file to evict from the cache. This value is compared to the number of blocks that are allocated to a file (KB_ALLOCATED), which might differ slightly from the file size.

Possible combinations of options are:

- Only Safe limit
- Safe limit + LRU
- Safe limit + SIZE
- Safe limit + FILENAME
- Safe limit + MINFILESIZE
- Safe limit + MAXFILESIZE
- Safe limit + LRU + FILENAME
- Safe limit + LRU + MINFILESIZE
- Safe limit + LRU + MAXFILESIZE
- Safe limit + SIZE + FILENAME
- Safe limit + SIZE + MINFILESIZE
- Safe limit + SIZE + MAXFILESIZE

--list-file *ListFile* – The specified file contains a list of files to be evicted, one file per line. All files must have fully qualified path names. File system quotas need not be specified. If the list of files to

be evicted have file names with special characters, then a policy must be used to generate the `listfile`. This policy output must be hand-edited to remove all other entries except the file names and can be passed to the `evict` command.

`--file FilePath` – The fully qualified name of the file to be evicted. File system quotas need not be specified.

enforceFilesetQuotaOnRoot – Enable to evict files created by root. By default, the files that are created by root are not evicted.

1. Evicting by using the filter option:

```
node1:/gpfs/cache/fileset_IW # mmlsfs fs1 -Q
flag value description
-----
Q user;group;fileset Quotas accounting enabled
none Quotas enforced
none Default quotas enabled
```

```
node1:/gpfs/cache/fileset_IW # mmquotaon -v fs1
```

```
mmquotaon on fs1
```

```
node1:/gpfs/cache/fileset_IW # mmlsfs fs1 -Q
```

```
flag value description
-----
Q user;group;fileset Quotas accounting enabled
user;group;fileset Quotas enforced
none Default quotas enabled
```

```
node1:/gpfs/cache/fileset_IW # mmlsquota -j fileset_IW fs1
```

Remarks	Block Limits File Limits											
	Filesystem type	KB quota	limit	in_doubt	grace	files quota	limit	in_doubt	grace			
	fs1	FILESET	96	102400	2097152	0	none	20	0	0	0	none

```
node1:/gpfs/cache/fileset_IW # mmadmctl fs1 evict -j fileset_IW --filter
FILENAME='%work%'
```

```
mmadmctl: Run mmcheckquota command before running mmadmctl with evict option
mmadmctl: Command failed. Examine previous error messages to determine cause.
```

```
node1:/gpfs/cache/fileset_IW # mmcheckquota fs1
```

```
fs1: Start quota check2 % complete on Thu Oct 27 09:19:41 2016
[...]
95 % complete on Thu Oct 27 09:20:03 2016
100 % complete on Thu Oct 27 09:20:03 2016
Finished scanning the inodes for fs1.
Merging results from scan.
```

```
node1:/gpfs/cache/fileset_IW # mmadmlocal ls workfile1
```

```
-rw-r--r-- 1 root root 104857600 Oct 27 08:30 workfile1
```

```
node1:/gpfs/cache/fileset_IW # mmafmctl fs1 evict -j fileset_IW --filter
FILENAME='%work%'
```

```
node1:/gpfs/cache/fileset_IW # mmafmlocal ls workfile1
mmafmlocal: Command failed. Examine previous error messages to determine cause.
```

2. Manual eviction by using the **--list-file** option:

```
[root@c21f2n08 ~]# ls -lshi /gpfs/fs1/evictCache
total 6.0M
27858308 1.0M -rw-r--r--. 1 root root 1.0M Feb  5 02:07 file1M
27858307 2.0M -rw-r--r--. 1 root root 2.0M Feb  5 02:07 file2M
27858306 3.0M -rw-r--r--. 1 root root 3.0M Feb  5 02:07 file3M
```

```
[root@c21f2n08 ~]# echo "RULE EXTERNAL LIST 'HomePREPDAEMON' RULE
'ListLargeFiles' LIST
```

```
'HomePREPDAEMON' WHERE PATH_NAME LIKE '%" > /tmp/evictionPolicy.pol
```

```
[root@c21f2n08 ~]# mmapplypolicy /gpfs/fs1/evictCache -I defer -P /tmp/
evictionPolicy.pol -f /tmp/evictionList
```

```
#Edited list of files to be evicted
```

```
[root@c21f2n08 ~]# cat /tmp/evictionList.list.HomePREPDAEMON
```

```
27858306 605742886 0  --/gpfs/fs1/evictCache/file3M
```

```
[root@c21f2n08 ~]# mmafmctl fs1 evict -j evictCache --list-file /tmp/
evictionList.list.HomePREPDAEMON
```

```
Evicted   (Total)   Failed
  1         (1)       0
tspcacheevict: List of failures : /var/mmfs/tmp/evictionFailedList.mmafmctl.23349
mmafmctl: Command failed. Examine previous error messages to determine cause.
```

```
[root@c21f2n08 ~]# ls -lshi /gpfs/fs1/evictCache
```

```
total 3.0M
27858308 1.0M -rw-r--r--. 1 root root 1.0M Feb  5 02:07 file1M
27858307 2.0M -rw-r--r--. 1 root root 2.0M Feb  5 02:07 file2M
27858306  0 -rw-r--r--. 1 root root 3.0M Feb  5 02:07 file3M
```

3. Manual eviction by using the **--file** option:

```
[root@c21f2n08 ~]# ls -lshi /gpfs/fs1/evictCache
total 3.0M
27858308 1.0M -rw-r--r--. 1 root root 1.0M Feb  5 02:07 file1M
27858307 2.0M -rw-r--r--. 1 root root 2.0M Feb  5 02:07 file2M
27858306  0 -rw-r--r--. 1 root root 3.0M Feb  5 02:07 file3M
```

```
[root@c21f2n08 ~]# mmafmctl fs1 evict -j evictCache --file /gpfs/fs1/
evictCache/file1M
```

```
[root@c21f2n08 ~]# ls -lshi /gpfs/fs1/evictCache/file1M
```

```
total 0
27858308 0 -rw-r--r--. 1 root root 1.0M Feb  5 02:07 file1M
```

4. Manual eviction of a partially cached file by using the **--file** option. In the following example, file1 is partially cached file of ro1 fileset. Out of 2MB, only 1MB data blocks are cached. You can use below steps to evict 1MB data blocks.

```
# ls -altrish /gpfs/fs1/ro1
```

```
total 2.0M
27858311 1.0M -rw-r--r--. 1 root root 2.0M Feb  5 02:07 file1
```

```
# mmadmctl fs1 evict -j ro1 --file /gpfs/fs1/ro1/file1
```

```
# ls -altrish /gpfs/fs1/ro1/file1
```

```
total 0
27858311 0 -rw-r--r--. 1 root root 2.0M Feb  5 02:07 file1M
```

Note: If a file is evicted from an AFM fileset, the snapshot file of the evicted file contains zeros.

Operation with disconnected home

With a cache and home cluster that is separated by a wide area network (WAN), it might result in intermittent outages and possibly long-term disruptions.

If the primary gateway determines that the home cannot be accessed, the primary gateway waits until the interval time specified in the **afmDisconnectTimeout** parameter passes, and then changes the cache state to disconnected. This feature is available on all AFM filesets.

In a disconnected state, cached files are served to applications from the cache. Application requests for uncached data return an I/O error. All update operations from the cache complete, and return successfully to the application. These requests remain queued at the gateway until they can be flushed to home.

In a disconnected state, the home cannot be accessed for revalidation. Therefore, the latest updates from home are not available in the cache. Writes, which require revalidation with home might appear temporarily stuck if it is done after the Home fails, and before the cache moves to disconnected state. With revalidation stuck waiting on the unavailable home, the request is timed out because of the value set in the **afmDisconnectTimeout** parameter.

If cache filesets are based on the NSD protocol, when the home file system is not mounted on the gateway, the cache cluster puts the cache filesets into unmounted state. These cache filesets never enter the disconnected state.

If the remote cluster on the home cluster does not respond due to a deadlock, operations that require remote mount access, such as revalidation or reading uncached contents, stop responding until the remote mount becomes available again. This remote cluster response is true for AFM filesets that use the NSD protocol to connect to the home cluster. You can continue accessing cached contents without disruption by temporarily disabling all of the revalidation intervals until the remote mount is accessible again.

If a cache fileset is disconnected for an extended period, the number of file system updates might exceed the buffering capacity of the gateway nodes. In this situation, operations continue in the cache. When the connection to home is restored, AFM runs recovery and synchronizes its local updates to the home cluster.

AFM automatically detects when home is available and moves the cache into the Active state. The **afmHomeDisconnected** callback event and the **afmHomeConnected** callback event can be used to monitor when a cache changes state.

Fileset that is using a mapping target go into disconnected state if the NFS server of the primary gateway is unreachable, even when the NFS servers of all participating gateways are reachable.

Prefetch tasks that fail due to home disconnection, continue when home is available again.

The following example shows the number of read/write operations that were run while the home was in the disconnected mode.

```
node1:/gpfs/cache/fileset_IW # mmadmctl fs1 getstate
Fileset Name Fileset Target Cache State Gateway Node Queue Length Queue numExec
-----
fileset_IW nfs://node4/gpfs/fshome/fset002new Disconnected node2 0 102
```

node1:/gpfs/cache/fileset_IW # ls -la

```
total 128
drwx----- 65535 root root 32768 Oct 13 16:50 .afm
-rw-r--r-- 1 root root 8 Oct 22 17:02 newfile2
drwx----- 65535 root root 32768 Oct 22 05:23 .pconflicts
drwx----- 65535 root root 32768 Oct 22 17:02 .ptrash
dr-xr-xr-x 2 root root 32768 Oct 13 16:20 .snapshots
```

node1:/gpfs/cache/fileset_IW # for i in 1 2 3 4 5 ; do date > file\$i ; done

node1:/gpfs/cache/fileset_IW # mmadmctl fs1 getstate

```
Fileset Name Fileset Target Cache State Gateway Node Queue Length Queue numExec
-----
fileset_IW nfs://node4/gpfs/fshome/fset002new Disconnected node2 27 102
```

turn on NFS on the home.

node1:/gpfs/cache/fileset_IW # mmadmctl fs1 getstate

```
Fileset Name Fileset Target Cache State Gateway Node Queue Length Queue numExec
-----
fileset_IW nfs://node4/gpfs/fshome/fset002new Dirty node2 27 102
```

node1:/gpfs/cache/fileset_IW # mmadmctl fs1 getstate

```
Fileset Name Fileset Target Cache State Gateway Node Queue Length Queue numExec
-----
fileset_IW nfs://node4/gpfs/fshome/fset002new Dirty node2 27 102
```

node1:/gpfs/cache/fileset_IW # date > file10

node1:/gpfs/cache/fileset_IW # mmadmctl fs1 getstate

```
Fileset Name Fileset Target Cache State Gateway Node Queue Length Queue numExec
-----
fileset_IW nfs://node4/gpfs/fshome/fset002new Active node2 0 134
```

Expiring a disconnected RO cache

Read-only (RO) filesets can be configured to expire cached data after the gateway nodes are in a disconnected state for a specified amount of time.

This feature provides control over how long a network outage between the cache and home can be tolerated before the data in the cache is considered stale. This control prevents access to old data, where old is defined by the amount of time that the WAN cache is out of synchronization with the data on the home site. Cached data is only available until the time period set in the **afmExpirationTimeout** parameter expires, at which point the cached data is considered 'expired' and cannot be read until the network is reconnected.

This feature is not available in other AFM modes.

RO cache filesets that are created by using the NSD protocol do not automatically go to the expired state, even if the expiration timeout is set for them. The RO cache filesets remain in the unmounted state until the home file system is unavailable. The administrator can manually expire the data in an RO cache before the expiration timeout is reached by using the **mmafmctl expire** command only if the expiration timeout is configured for the fileset. An RO cache that is created by using NSD protocol can also be manually expired. Use the following command:

```
#mmafmctl Device {resync | expire | unexpire | stop | start} -j FilesetName
```

For more information, see *mmafmctl* command in *IBM Spectrum Scale: Command and Programming Reference*.

1. Expiring an RO fileset -

```
# mmafmctl fs1 expire -j ro1
```

```
# mmafmctl fs1 getstate -j ro1
```

Fileset Name	Fileset	TargetCache	State	Gateway	Node	Queue	Length	Queue	numExec
ro1	gpfs:///gpfs/remotefs1/dir1		Expired			c26c4apv1	0		4

2. Unexpiring an RO fileset -

```
# mmafmctl fs1 unexpire -j ro1
```

```
#mmafmctl fs1 getstate -j ro1
```

numExec	Fileset Name	Fileset	Target	Cache	State	Gateway	Node	Queue	Length	Queue
0	ro1	gpfs:///gpfs/remotefs1/dir1			Active			c26c4apv1		

The **afmFilesetExpired** callback event is triggered when a fileset moves into expired state. The **afmFilesetUnexpired** callback event is triggered when a fileset moves out of the expired state.

Viewing snapshots at home

All snapshots at home for RO and LU filesets can be viewed in the cache by setting the **afmShowHomeSnapshot** parameter to yes. This variable is not applicable for SW and IW filesets.

The home and cache can have different snapshot directories for clarity. If the **afmShowHomeSnapshot** parameter is changed after fileset creation, the change is reflected in the cache only after IBM Spectrum Scale is restarted, or the file system is remounted. However, if you change the **afmShowHomeSnapshot** parameter value from yes to no during the lifetime of a fileset, it continues to show the home snapshots even after a restart, or file system remount.

For RO and LU filesets that use NFS, when a lookup is performed from the parent directory of . snapshot directory, the changes on the home are not reflected in the cache, as NFS cannot detect the changes in . snapshot. The **mtime/ctime** parameter of this directory is not modified on the home despite snapshots creates or deletes operations on the home. This is not true for RO and LU fileset that use GPFS backend as the latest snapshot directory gets reflected in cache.

AFM cannot detect a snapshot path on a home site that is not the IBM Spectrum Scale home site. To detect the snapshot path on this IBM Spectrum Scale or GPFS home when data is being migrated and prefetched, set the **afmShowHomeSnapshot** parameter. Because of this parameter, AFM detects the snapshot path on the home and does not pull snapshot files during migration.

You can set this parameter at a cluster level. Set this parameter if a home site is not the IBM Spectrum Scale site. On an IBM Spectrum Scale or GPFS home, you need not to set this parameter because AFM automatically detects the home snapshot path.

Changing home of AFM cache

AFM filesets continue to function though failures are occurred on a home.

AFM filesets serve the cache applications with cached data. If the home is permanently lost, you can create a new home. An existing SW/IW AFM cache can make the following changes to the home. These changes are not supported on RO/LU filesets.

1. Replace the home with a new empty home where the new target is created by using the NFS or NSD mapping:

The administrator must run the **mmafmctl failover** command without **--target-only** option to point to the new home. The new home is expected to be empty. To ensure that extended attributes are synchronized, run the **mmafmconfig** command on the new home before you run the failover command. If the new target is a mapping, failover does not split data transfers and queues them as normal requests without parallel data transfer.

2. Replace any of the following on an existing home:

- a. Replace the communication protocol (NSD, NFS): The administrator must run the **mmafmctl failover** command without the **--target-only** option, by using the new target protocol.

Note: Only the protocol changes. The home path does not change.

- b. Enable or disable parallel data transfer by shifting between NFS and a mapping: The administrator must run the **mmafmctl failover** command without the **--target-only** option, by using the new target protocol or mapping.

Note: Only the protocol changes. The home path does not change.

- c. Replace either the IP address or the NFS server by using the same communication protocol and home path: The administrator must run the **mmafmctl failover** command with the **--target-only** option. The IP/NFS server must be on the same home cluster and must be of the same architecture as the old NFS server.

Note: Only the IP or NFS server changes.

During failover, ensure that the cache file system is mounted on all gateway nodes, and the new home file system is mounted on all the nodes in the home cluster.

Note: Failover does not use parallel data transfers.

When you create a new home, all cached data and metadata available in the cache are queued in the priority queue to the new home during the failover process. The failover process is not synchronous and completes in the background. The **afmManualResyncComplete** callback event is triggered when failover is complete. Resync does not split data transfers even if parallel data transfer is configured, and the target is a mapping.

If the failover is interrupted due to a gateway node failure or quorum loss, failover is restarted automatically when the cache fileset attempts to go to Active state.

When a cache has multiple IW filesets, the administrator must choose a primary IW cache and fail this cache over to a new empty home. All of the other IW cache filesets to the old home must be deleted and re-created. If another IW cache is failed over to the same home after it is failed over to another IW cache, all the data in that cache overwrites existing objects at home. For this reason, fail over to a non-empty home is discouraged.

When the failover function is likely to be used due to the failure of the old home, the admin must be cautious and disable automatic eviction. You need to ensure that the eviction does not free the cached data on the cache. The evicted data cannot be recovered if the old home is lost.

Each AFM fileset is independently managed and has a one-to-one relationship with a target, thus allowing different protocol backends to coexist on separate filesets in the same file system. However, AFM does not validate the target for correctness when a fileset is created. The user must specify a valid target. Do not use a target that belongs to the same file system as the AFM fileset. For more information, see **mmafmctl** command in *IBM Spectrum Scale: Command and Programming Reference*.

The following example shows changing the target for an SW fileset. Consider a `fileset_SW` SW fileset of a `fs1` file system that uses the `nfs://node4/gpfs/fshome/fset001` home target. A failover is performed to a new target `nfs://node4/gpfs/fshome/fset002new`.

```
# mmlsfileset fs1 fileset_SW --afm
```

A sample output is as follows:

Filesets in file system 'fs1':			
Name	Status	Path	afmTarget
fileset_SW	Linked	/gpfs/cache/fileset_SW	nfs://node4/gpfs/fshome/fset001

```
# mmafmctl fs1 getstate -j fileset_SW
```

A sample output is as follows:

Fileset Name	Fileset Target	Cache State	Gateway Node	Queue Length	Queue numExec
fileset_SW	nfs://node4/gpfs/fshome/fset001	Active	GatewayNode1	0	6

```
# mmafmctl fs1 failover -j fileset_SW --new-target nfs://node4/gpfs/fshome/fset002new
```

A sample output is as follows:

```
mmafmctl:Performing failover to nfs://node4/gpfs/fshome/fset002new
Fileset fileset_SW changed.
mmafmctl: Failover in progress. This may take while...
Check fileset state or register for callback to know the completion status.
```

```
# mmafmctl fs1 getstate -j fileset_SW
```

A sample output is as follows:

Fileset Name	Fileset Target	Cache State	Gateway Node	Queue Length	Queue numExec
fileset_SW	nfs://node4/gpfs/fshome/fset002new	NeedsResync	GatewayNode1	6	0

```
# mmafmctl fs1 getstate -j fileset_SW
```

A sample output is as follows:

Fileset Name	Fileset Target	Cache State	Gateway Node	Queue Length	Queue numExec
fileset_SW	nfs://node4/gpfs/fshome/fset002new	Recovery	GatewayNode1	6	0

```
# mmafmctl fs1 getstate -j fileset_SW
```

A sample output is as follows:

Fileset Name	Fileset Target	Cache State	Gateway Node	Queue Length	Queue numExec
fileset_SW	nfs://node4/gpfs/fshome/fset002new	Active	GatewayNode1	0 6	0

Note: After failover, the fileset changes to states such as `NeedsResync` or `Recovery`. Depending on the size or the elapsed time, the fileset remains in these transition states before the fileset turns into the `Active` state. The failover process is complete after the fileset is in `Active` state.

```
# mmlsfileset fs1 fileset_SW --afm
```

A sample output is as follows:

Filesets in file system 'fs1':			
Fileset Name	Status	Path	afmTarget
fileset_SW	Linked	/gpfs/cache/fileset_SW	nfs://node4/gpfs/fshome/fset002new

Out-band Failover - You can choose to copy all cached data offline from the AFM cache to the new home with any tool that preserves modification time (**mtime**) with nanoseconds granularity. An example of such a tool is - rsync version 3.1.0 or later with protocol version 31. After the data is copied, you can run the **mmafmctl failover** command to compare **mtime** and **filesize** at home, and avoid queuing unnecessary data to home.

Resync on SW filesets

A conflict situation might arise due to an inadvertent write on an SW home. AFM runs a resynchronization automatically.

Writing data on an SW home is not allowed under normal circumstances. However, AFM does not enforce this condition. If an inadvertent write or corruption occurs on an SW home due to some error situation, it results in a conflict scenario for the single-writer fileset.

Note: Resync does not use parallel data transfers. Even if parallel data transfer is configured and the target is a mapping, the data transfers are not split.

If AFM detects inconsistencies during flushing, the fileset goes into NeedsResync state. When the fileset is in the NeedsResync state, AFM runs a resynchronization automatically during the next request to the primary gateway, and the inconsistencies are resolved.

If resync is not triggered automatically, the inconsistencies must be manually resolved by running the **resync** command. When you run a resync, the fileset must be in the Active or the NeedsResync state. Resync is not allowed on IW filesets as multiple cache filesets can update the same set of data. Use the following command - **mmafmctl Device {resync | expire | unexpire} -j FilesetName**. For more information, see *mmafmctl* command in *IBM Spectrum Scale: Command and Programming Reference*.

mmafmctl fs1 resync -j sw1

```
# mmafmctl fs1 getstate -j sw1
Fileset  Fileset Target      Cache  Gateway  Queue  Queue
Name                                           State  Node    Length numExec
-----
sw1      nfs://c26c3apv2/gpfs/homefs1/newdir1  Dirty  c26c2apv1  4067   10844
```

During a resync, all cached data and metadata are queued in the priority queue. The resynchronization is an asynchronous process and completes in the background. The `afmManualResyncComplete` callback event is triggered when the resynchronization is complete. When the complete priority queue is flushed, the fileset state changes to Active.

Evicted files are not synchronized to home.

A resync of partially cached files pushes only the cached data blocks to the new home. Uncached blocks are filled with null bytes.

Important: Do not use on an SW fileset, except when it is required to correct the home under such conflict scenarios.

Out-band Resync - You can choose to copy all cached data offline from the AFM cache to the new home with any tool that preserves modification time (**mtime**) with nanoseconds granularity. An example of such a tool is - rsync version 3.1.0 or later with protocol version 31. After the data is copied, you can run the **mmafmctl failover** command to compare **mtime** and **filesize** at home, and avoid queuing unnecessary data to home.

The Resync operation does not delete new files or directories that are created on the home nor new names that are given to files or directories at the home.

In some cases, a cache has pending changes such as delete and rename to replicate and the fileset recovery is triggered. In this case, the home might have old data and renamed files or directories are replicated to the home. Some extra files might exist at home because of the files or directories replication. The cache and the home have the latest copy of the files.

Note: When the **afmResyncVer2** parameter is enabled on an SW fileset or a primary fileset, the extra files are deleted. The resync operation is run on a cache fileset to ensure that the cache and the home are synchronized.

AFM resync version 2

The AFM resync version 2 enhances the replication performance. This feature is suitable for a cluster that is employed for heavy stress and workloads. It uses an on-demand dependency resolution for queued messages, in case of resync and/or recovery is running. The message queuing performance is increased by using the runtime filtering or the dependency resolution. Thus, less memory is used on the gateway nodes and queued messages are replicated quickly. This feature also helps in the role reversal of AFM-DR filesets.

An AFM and AFM-DR primary fileset creates a file operation queue on the designated gateway nodes for a replication. These operations are run on the target or home cluster asynchronously based on the setting of the **afmAsyncDelay** parameter. For any dependent operations in the queue, this asynchronous delay helps to filter operations that need not to be sent to the target over an underlying protocol. This filtering of operations saves the bandwidth. For more information, see [“Asynchronous delay” on page 48](#).

When many incoming operations are generated because of heavy workloads, the queue length increases. On the gateway node, messages or operations are stored in the memory and processed for execution, filtering, or queued based on an operation type. For heavy workloads, the memory usage on nodes affects the system performance and the network throughput between cache and home filesets or primary and secondary filesets. When any gateway node fails, you must run demanding recovery and/or resync operations on a new gateway node. When the **afmResyncVer2** parameter is enabled queued messages are replicated based on the on-demand dependency resolution to the home or secondary whenever the queued messages are ready. The execution of queue to the home or secondary does not pause on certain dependent operations because the memory is available for more operations after the queues are replicated.

A single-writer (SW) cache fileset or an AFM-DR primary fileset can be configured for the AFM resync version 2 by setting the **afmResyncVer2** parameter in the **mmchfileset** command.

To set or unset the **afmResyncVer2** parameter on an AFM or AFM-DR fileset, you need to stop or unlink the fileset. For more information, see [“Stop and start replication on a fileset” on page 90](#).

AFM internally stores all the information necessary to replay the updates that are made in the cache to the home cluster. When a gateway node fails, the in-memory queue for any hosted fileset is lost. Any filesets that were hosted on the failed gateway nodes are transferred to another gateway nodes. The new gateway nodes rebuild queues in the memory. When a gateway node fails, the fileset that is hosted on the gateway node is transferred to another gateway node. This gateway node with the fileset builds the queue in the memory. This process is called recovery. During the recovery, outstanding cache updates are placed in the in-memory queue and the gateway processes the queue. AFM collects the pending operations by running a policy scan on the fileset. AFM uses the policy infrastructure in IBM Spectrum Scale to engage all the nodes that are mounting the file system to participate in the scan process. Pending requests, which are discovered by the recovery process, are queued in a special queue called the priority queue. At the same time, a normal queue is also created. The normal queue is used for new incoming operations to the fileset.

When the **afmResyncVer2** parameter is enabled, messages in the queue are checked for dependent operations. An example of dependent operations that need to be performed is as follows:

1. Create a file.
2. Change the file name.
3. Add some data the file.

However, there are other operations that can be run whenever they are available in the queue because they are not dependent on any other operations. The memory of a gateway node is saved where messages are queued.

Enable the AFM resync version 2 feature by using the following steps:

1. Verify the primary fileset information.

```
# mmlsfileset fs1 pri --afm -L
```

The sample output is as follows:

```
Filesets in file system 'fs1':

Attributes for fileset pri:
=====
Status                               Linked
Path                               /gpfs/fs1/pri
Id                                  1
Root inode                          524291
Parent Id                           0
Created                             Thu Feb 11 15:05:23 2021
Comment
Inode space                          1
Maximum number of inodes            100352
Allocated inodes                    100352
Permission change flag              chmodAndSetacl
afm-associated                       Yes
Target                             nfs://c7f2n06/gpfs/fs1/sec
Mode                                primary
Async Delay                         15 (default)
Recovery Point Objective             disable (default)
Last pSnapId                         1
Number of Gateway Flush Threads     4
Primary Id                          2836795238842262449-C0A8693E60255310-1
IO Flags                            0x0 (default)
```

2. Stop the primary fileset.

```
# mmafmctl fs1 stop -j pri
```

3. Enable the AFM resync version 2 feature.

```
# mmchfileset fs1 pri -p afmResyncVer2=yes
```

4. Start the primary fileset.

```
# mmafmctl fs1 start -j pri
```

5. Verify the primary fileset information again.

```
# mmlsfileset fs1 pri --afm -L
```

The sample output is as follows:

```
Filesets in file system 'fs1':

Attributes for fileset pri:
=====
Status                               Linked
Path                               /gpfs/fs1/pri
Id                                  1
Root inode                          524291
Parent Id                           0
Created                             Thu Feb 11 15:05:23 2021
Comment
Inode space                          1
Maximum number of inodes            100352
Allocated inodes                    100352
Permission change flag              chmodAndSetacl
afm-associated                       Yes
Target                             nfs://c7f2n06/gpfs/fs1/sec
Mode                                primary
Async Delay                         15 (default)
Recovery Point Objective             disable (default)
Last pSnapId                         1
Number of Gateway Flush Threads     4
Primary Id                          2836795238842262449-C0A8693E60255310-1
IO Flags                            0x10000 (afmResyncVer2)
```

Using IBM Spectrum Protect for Space Management

IBM Spectrum Protect for Space Management can be used on the AFM filesets or on the home.

The following figure illustrates IBM Spectrum Protect for Space Management connected to home.

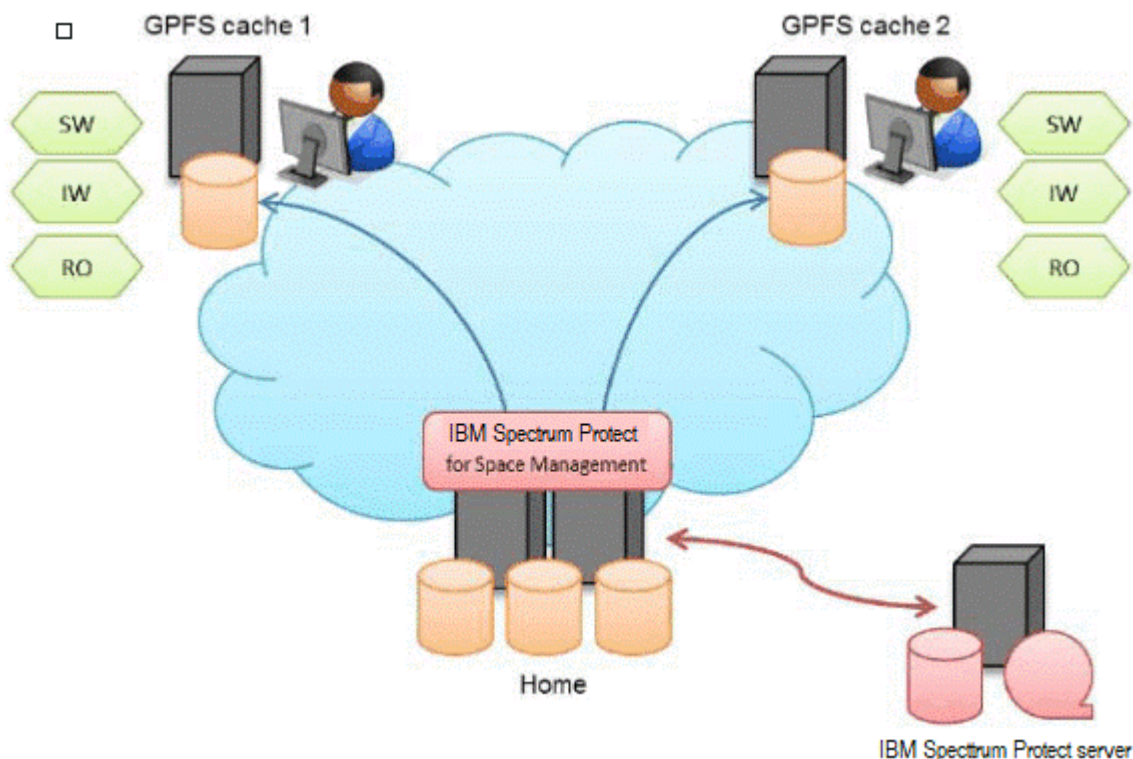


Figure 11. Sample setup of IBM Spectrum Protect for Space Management connected to home

A new file created at home becomes candidate for migration to the IBM Spectrum Protect server. When a migrated file at home is read from the cache, the file is recalled at home and served to the cache. If the cache does a write on a migrated file at home, the file is recalled at home and written to when the AFM queue is flushed. If multiple migrated files are written at the same time, the home issues recalls for all files at the same time. It is recommended that you exclude frequently changing files from IBM Spectrum Protect for Space Management migration process to avoid recalls.

The following figure illustrates IBM Spectrum Protect for Space Management connected to both home and cache.

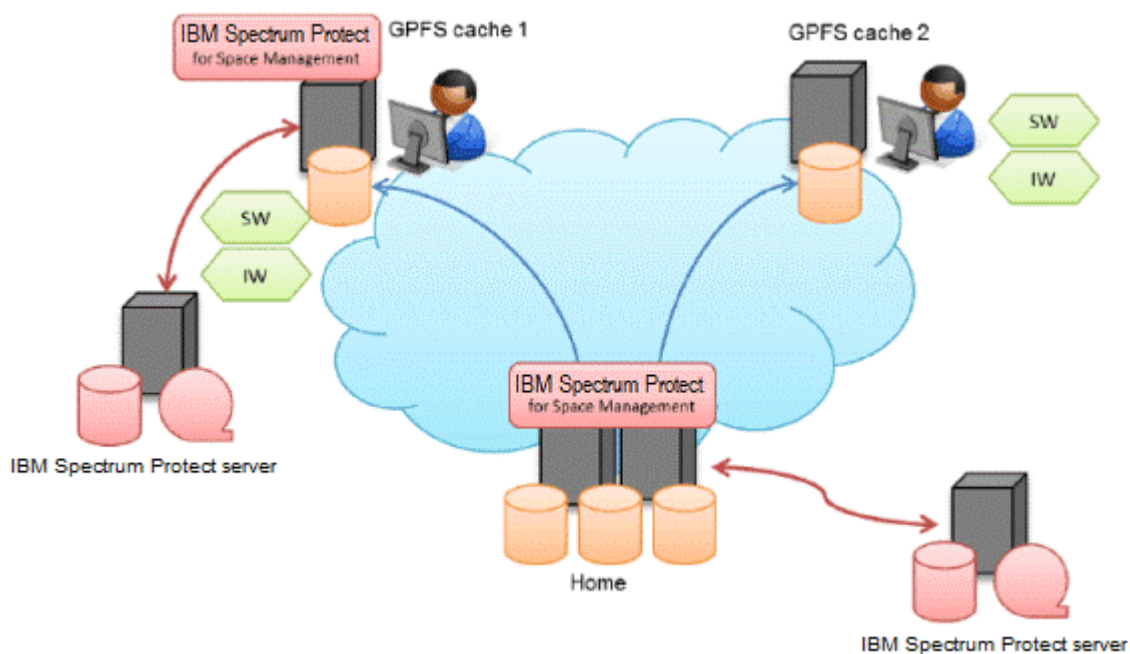


Figure 12. IBM Spectrum Protect for Space Management connected to both home and cache

When using IBM Spectrum Protect for Space Management on an AFM fileset, the flag **AFMSKIPUNCACHEDFILES** must be set in the `dsm.sys` configuration file (IBM Spectrum Protect-related) to yes. For example - **AFMSKIPUNCACHEDFILES** yes. This parameter should be used for read-write cache filesets. It prevents the migration of dirty and uncached files. If this flag is not set, it might result in long waiters or unexpected cache states and unexpected errors in terms of the IBM Spectrum Protect for Space Management migration processing. In the LU mode when this parameter is set and a file is made local because of updating data, migration of the file to tape might be prevented. For migration in the LU mode, do not unset this parameter. Migrated files cannot be evicted. File operations such as read, write, truncate, append or rename on migrated files recalls the files into the cache. When multiple migrated files are modified at the same time all recalls are submitted at the same time, IBM recommends that you exclude frequently changing files or frequently read files from IBM Spectrum Protect for Space Management migration process on both cache and home to avoid recalls.

It is recommended the following guidelines while using IBM Spectrum Scale AFM and IBM Spectrum Protect:

- Prevent cache eviction in combination with IBM Spectrum Protect on the same fileset. Both techniques have the same goal to reduce the space required in the fileset. The combination of both techniques unnecessarily increases the complexity of the environment.
- IBM Spectrum Scale snapshots and IBM Spectrum Protect have a limited compatibility. The deletion of a stub file that is reflected in a snapshot (the snapshot was generated before or after the file was migrated) causes the recall of the file data. The file data is stored in the snapshot so that it can be accessed later. Therefore, do not use snapshots for an AFM fileset (in home or cache) and in the file system hosting the AFM fileset, if you are using IBM Spectrum Protect for Space Management.
- When using IBM Spectrum Protect on home or cache be aware that access (read or write) to multiple migrated files at the same time causes bulk recalls. Access to multiple files can be caused by users such as when they copy an entire directory or by AFM when changed files are replicated to home where the previous versions are migrated. You can avoid these issues by using the optimized tape recall process, which requires a list of files to be recalled before processing.

When running IBM Spectrum Scale AFM and IBM Spectrum Protect backup operations, prevent cache eviction in combination with IBM Spectrum Protect backup on the same fileset, if possible. Evicted (uncached) files will be skipped from backup processing. This might lead to errors in terms of the versioning of files on the IBM Spectrum Protect server.

For detailed description about the setup and configuration of IBM Spectrum Protect for Space Management for AFM, see [Configuring IBM Spectrum Scale Active File Management](#).

Performing a planned maintenance by using the IW cache

You can perform a planned maintenance by using IW cache.

Let us assume that we have IW cache on Side A that is hosting applications. IW cache points to a home in Side B. Complete the following steps for a planned maintenance of Side A:

1. Stop all applications in IW cache (Side A).
2. Flush the pending queue on IW cache (Side A) to home (Side B) by using **mmafmctl Device flushPending -j FileSetName**.
3. Unlink IW cache (Side A).
4. Start applications at home (Side B).
5. Relink the cache after maintenance window.
6. Stop the applications at home (Side B).
7. Start the applications at cache (Side A).
8. Schedule the background prefetch if required (Side A).

The following figure illustrates IW cache (Side A) to home (Side B).

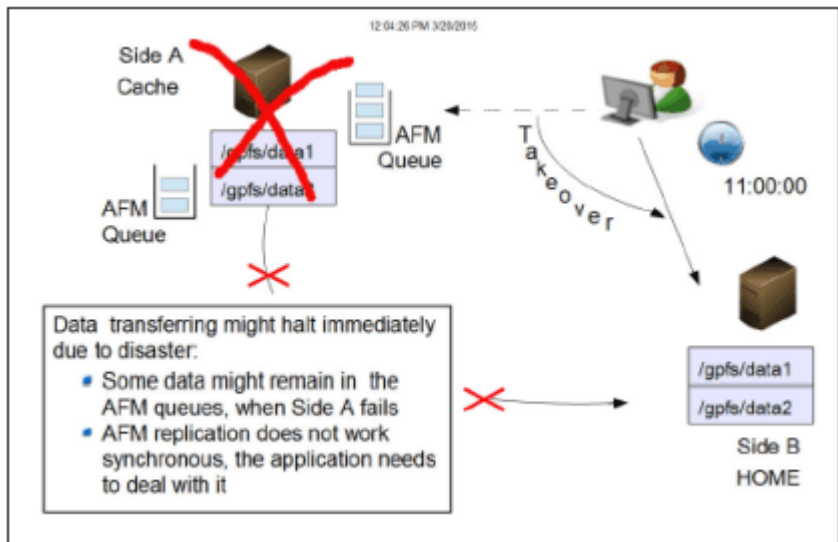


Figure 13. IW cache (Side A) to home (Side B)

Handling IW cache disaster

You can perform planned or unplanned maintenance by using the IW cache.

Performing an unplanned maintenance using IW cache

You can perform an unplanned maintenance by using the IW cache.

For an unplanned outage, some changes might not be synchronized with the home. IW can handle requests that are lost from the cache due to a disaster, while allowing application updates at the home to the same data.

The cache became unavailable because of the disaster at the IW cache. Therefore, the IW cache is down with pending updates (Side A). Steps to follow in an unplanned outage are:

1. Start applications at home (Side B). Record the time.
2. When the IW cache site (Side A) is ready to take over again, stop applications at the home site (Side B).

The following figure illustrates IW cache site (Side A) to home site (Side B).

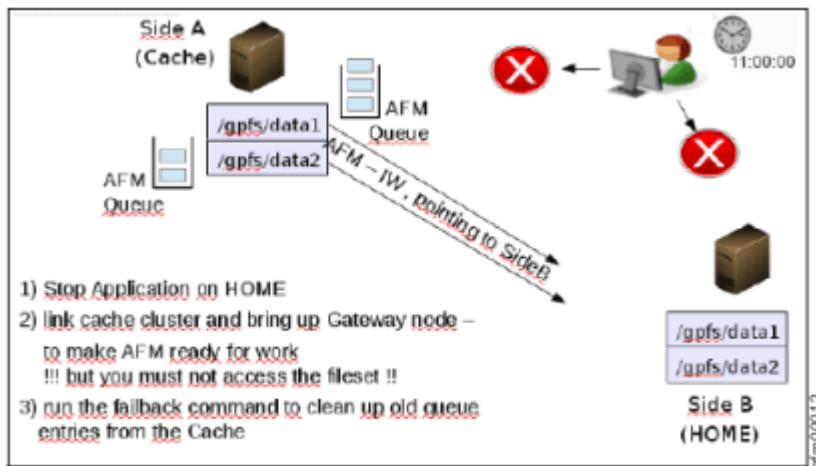


Figure 14. IW cache site (Side A) and home site (Side B)

3. Link the cache site back (Side A) and ensure that the gateways are up. Do not access the fileset directories.

If the cache directories are accessed, recovery is triggered. Old contents from the cache, which were pending, are synchronized with the home at the end of recovery, over-writing latest application changes at home. Perform the following steps so that the latest data is preserved and old or stale data is discarded. The data staleness is determined based on file mtime and failover time. The failover time is the time when applications were moved to the home, given as input to the **failback** command.

4. Run the following command from the cache site to synchronize the latest data, specifying the time when the home site became functional and including the complete time zone, in case home and cache are in different time zones:

```
mmafmctl FileSystem failback -j Fileset --start --failover-time  
'TimeIncludingTimezone'
```

The failback command resolves conflicts between pending updates in cache at the time of failover with the data changes at home. When the synchronization is in progress, the fileset state is **FailbackInProgress**. The fileset is read-only when the failback is in progress.

After failback completes, the fileset state is **FailbackCompleted**. The failback process resolves conflicts in the following way:

- a. Dirty data that was not synchronized is identified.
- b. If the changed files or directories were not modified at the home when the applications were connected to the home, the cache pushes the change to the home to avoid any data loss.
- c. If the files or directories are recently modified at the home, the cache discards the earlier updates from the cache. The next lookup on the cache brings the latest metadata from the home.

If the conflicted dirty data is a directory, it is moved to **.ptrash**. If the conflicted dirty data is a file, it is moved to **.pconflict**. The administrator must clean **.pconflict** directory regularly. If IW is converted to the other modes, **.ptrash** and **.pconflict** directories remain.

5. After achieving the **FailbackCompleted** state, run the following command to move the fileset from the Read-Only to the Active, where it is ready for use. If the command is not run successfully, run the command again.

```
mmafmctl FileSystem failback -j Fileset --stop
```

Note: If failback is not complete or if the fileset moves to **NeedFailback**, run the failback command again.

The cache site is ready for use. All applications can start functioning from the cache site. New files that are created at home are reflected in the cache on the next access, based on the revalidation interval. Failback does not pull in data of uncached files from home, which needs to be done explicitly by the administrator by using the **mmafmctl prefetch** command. If failback might be used, the

ctime of RENAME operations in the home file system must be updated. This is enabled by using **setCtimeOnFileRename** at home:

```
mmchconfig setCtimeOnFileRename=yes -i
```

Note: Failback does not work if revalidation is disabled in the IW cache site.

Using AFM with encryption

AFM supports file encryption. Encryption can be applied to AFM-managed filesets.

AFM home sites and cache sites can be enabled with encryption, independent of each other. The data is encrypted while at rest (on disk) and is decrypted on the way to the reader or application that is hosted on home and caches. However, AFM communication between home and cache is not encrypted.

With the data that is flowing between home and cache filesets not being encrypted by the adoption of file encryption. The communication between the clusters needs to be encrypted explicitly (if the privacy of the data over the network is a concern) by ensuring that a cipher list is configured. To ensure that the data is transmitted in the encrypted form, a cipher other than AUTHONLY must be adopted. AES128-GCM-SHA256 is one of the recommended ciphers. Run the **mmauth show** command to view the cipher lists used to communicate within the local and with the remote clusters. To ensure that all file content on disk and on the network is encrypted, configure file encryption at home and on the caches. Also, configure a cipher list on all the clusters, ensuring that ciphers are configured within and across clusters. Because of the file encryption, the data is transmitted in the encrypted form between NSD clients and servers (both directions). However, the file metadata or RPC headers are encrypted. Only the use of encrypted communications (cipher list) ensures that the entire message content gets encrypted.

If the NFS protocol is used for communication between the home and cache clusters, and privacy of the data over the network is a concern, then to encrypt the NFS transfers you need to set up an encrypted tunnel. If a cluster configured as cache includes an encrypted file system, then all nodes in the cluster, and especially the gateway nodes, require access to the Master Encryption Keys.

For encryption setup, see *Encryption* in the *IBM Spectrum Scale: Administration Guide*.

Using mmbackup

When the **mmbackup** command is run on a file system that has AFM filesets, only the cached data from the filesets is backed up.

The **mmbackup** command does not back up the uncached data from the AFM filesets.

In a file system that is IBM Spectrum Protect for Space Management managed, a backup operation in either the cache or home fileset skips migrated offline files that were not yet backed up. The backup of a file that is evicted from a cache causes the file to be read again from home to cache. To avoid reading the file again when eviction on the cache site is enabled, ensure that the backup occurs before the eviction.

Disabling AFM

An AFM fileset can be disabled and converted to a GPFS-independent fileset. However, GPFS-independent fileset cannot be converted to an AFM fileset.

To dissociate home and cache relationship of an AFM fileset, you can convert the AFM fileset to an independent fileset by using the **mmchfileset -p afmTarget=disable** command. Thereafter, the fileset is a GPFS or IBM Spectrum Scale-independent fileset and requests are not queued to home. An example is the case of NFS migration, where AFM fileset must be disabled after migration.

You must disable AFM filesets while the filesets are linked. In some cases, disabling of AFM fileset might fail with an error message.

Complete the following steps to disable an AFM fileset -

1. If the fileset is unlinked, run **mmmlinkfileset gpfs0 FilesetName -J gpfs0/FilesetName** to link the fileset.
2. Run **mmchfileset gpfs0 FilesetName -p afmTarget=disable**.

The following warning message displays - Warning! Once disabled, AFM cannot be re-enabled on this fileset. Do you wish to continue? (yes/no)

Enter yes.

The following warning message displays - Warning! Fileset should be verified for uncached files and orphans. If already verified, then skip this step. Do you wish to verify same? (yes/no)

Enter yes.

- If error conditions are not found, the following message displays:

File system `gpfs0` is mounted on *n* nodes or fileset *FilesetName* is not unlinked.

Unlink the fileset and run **mmchfileset gpfs0 FilesetName -p afmTarget=disable** again. Enter no to the warning message about verification of the fileset. The AFM fileset is disabled.

- In case of error conditions, disabling AFM fileset fails. See the following table for the error conditions with the corresponding messages.

Table 7. Conditions in which disabling AFM fileset fails, with corresponding messages	
Condition	Error message
Incomplete directories present	mmchfileset: 6027-2307 [E] Uncached files present, run prefetch first.
Uncached files present	Uncached files present, run prefetch first using policy output: <code>/var/mmfs/tmp/cmdTmpDir.mmchfileset.18149/list-file.mmchfileset.18149</code>
Orphans present	Orphans are present, run prefetch first using policy output: <code>/var/mmfs/tmp/cmdTmpDir.mmchfileset.18149/list-file.mmchfileset.18149</code>

You must run prefetch as suggested in the message to fix the error condition.

3. If uncached files or orphans are present, run prefetch by using the following command:

```
mmafmctl gpfs0 prefetch -j FilesetName --list-file /  
var/mmfs/tmp/cmdTmpDir.mmchfileset.18149/list-file.mmchfileset.18149
```

4.

5. If incomplete directories are present, generate a list file to prefetch the uncached files first. Use this generated list file that contains uncached file entries to run the command - **mmafmctl gpfs0 prefetch -j FilesetName --list-file list file path**. Here, *list file path* is the generated list file.

6. Ensure that all data in the AFM fileset is prefetched.

7. Run step “2” on page 89 again and complete disabling AFM fileset.

Stop and start replication on a fileset

The stop and start operations can be used during the downtime planning of the replication activities on an AFM or AFM-DR fileset. When you stop the replication on an AFM fileset, the fileset state changes to 'Stopped' and the replication between both sites stops until you perform the 'start' operation on the stopped fileset. These operations are useful to control the replication during the downtime and manage the bandwidth efficiently.

When the fileset state is 'Stopped', any data modification at the cache or the home is not synchronized. That is, the changed data at the cache does not reflect in filesets at the home, and the home data is not available at the configured cache.

After the downtime, you need to perform the 'start' operation to start the replication activities on the fileset. The start operation first triggers the 'recovery' phase on the fileset, which determines the local changes and prepares the changed list for the replication. This list is sent to the assigned gateway node to synchronize the data with the home.

During the recovery phase, all pending operations are synchronized by using the Priority Queue on the gateway node. AFM or AFM-DR replicates a full file again if the file has any changes during the recovery phase because AFM does not track changed data at the file level. After the recovery phase is complete and all changes are synchronized to the home, the AFM fileset state changes to 'Active'.

If AFM-DR primary fileset is not synchronized to the secondary fileset and the primary fileset is in 'PrimInitInProg' during the maintenance, AFM queues all files again and synchronizes the queued files to the AFM-DR secondary. When the primary is synchronized with the secondary and 'psnap0' is created at both sites, AFM changes the primary fileset state and leaves the 'PrimInitInProg' state.

To stop replication of an AFM fileset, issue the following command:

```
#mmafmctl <fs> stop -j <fileset>
```

You can check the fileset state by using the following command:

```
#mmafmctl <fs> getstate -j <fileset>
```

A sample output is as follows:

Fileset Name	Fileset Target	Cache State	Gateway Node	Queue Length	Queue numExec
afmFileset	nfs://p7fbn09/gpfs/fs0/swfileset3	Stopped			

The state of the fileset is now 'Stopped'.

To start replication on a specific fileset, issue the following command:

```
#mmafmctl <fs> start -j <fileset>
```

If you perform other operations like stop or flushpending on a fileset that is in the 'Stopped' state, the system displays an error message as follows:

```
Operation not permitted
mmafmctl: [E] Operation is not permitted as state of the fileset is Stopped.
mmafmctl: Command failed. Examine previous error messages to determine cause.
```

A similar message like this error message is displayed, when you perform the start operation on a fileset that is not in the 'Stopped' state.

AFM IPv6 Support

Internet Protocol Version 6 (IPv6) is the protocol that is designed by the IETF to replace Internet Protocol Version 4 (IPv4). Active File Management (AFM) file set now supports connectivity and administration over IPv6.

AFM, AFM-based Asynchronous Disaster Recovery (AFM DR), and AFM to cloud object storage files set supports connectivity and administration over IPv6. The Spectrum Scale cluster can be configured to use its daemon and admin communication over IPv6 addresses.

AFM gateway nodes and mapping support IPv6 address as well. After the IBM Spectrum Scale cluster is set up for using IPv6 as its network, filesets can be created by using these IPv6 addresses. Also, the filesets can be converted to use IPv6 addresses by using **mmafmctl failover --target-only** option and AFM DR filesets can be converted to use IPv6 by using **mmafmctl ChangeSecondary --target-only** option.

To set up IPv6 support for IBM Spectrum Scale cluster, see *Enabling a cluster for IPv6* in the *IBM Spectrum Scale: Administration Guide*.

To change IP addresses or hostnames of cluster nodes, see *Changing IP addresses or host names of cluster nodes* in the *IBM Spectrum Scale: Administration Guide*.

Important: The IPv6 addresses must be mentioned in the square brackets when the IPv6 is configured with the AFM-related commands.

Examples

1. Setting remote cluster AFM target as IPv6.

```
mmcrfileset fs2 drp-fset-2 --inode-space=new -p afmtarget=[2001:192::321:5eff:febf:a69b]:/
gpfs/fs1/drs-fset-2 -p afmmode=primary
```

The output looks similar to the following as shown:

```
Fileset drp-fset-2 created with id 258 root inode 63963139.
Primary Id (afmPrimaryId) 134157187695697640-C0A876075E27E51D-258
```

2. Verifying fileset status.

```
mmafmcctl fs1 getstate
```

The output looks similar to the following as shown:

Fileset Name	Fileset Target	Cache State	Gateway
Node Queue Length Queue numExec			
obj1	https://s3.amazonaws.com:443/mufileset	Active	
c7f2n04	0 143119		
ip1	nfs://[2001:192::310:18ff:fec6:ecd8]/fileset	Active	
c7f2n03	0 1167977		
ip2	nfs://[2001:192::310:18ff:fec6:ecd8]/fileset1	Active	
c7f2n03	0 524752		

Note: To verify whether the hostname resolves to an IPv6 addresses, use `mmcml host <hostname>` command.

Example of changing the target address from IPv4 to IPv6 by using failover

Use the following details for changing the target address from IPv4 to IPv6 by using failover.

After the IBM Spectrum Scale cluster is enabled with IPv6 addresses, following steps can be used to failover existing filesets that are connected to the remote cluster over IPv4 to IPv6 addresses.

Note: Remote IBM Spectrum Scale cluster is also enabled with IPv6 addresses.

Cache cluster

```
mmiscluster
```

```
GPFS cluster information
```

```
=====
```

```
GPFS cluster name:      pry.Node2
GPFS cluster id:        2836795238850206208
GPFS UID domain:        pry.Node2
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	Node2	2001:192::42f2:e9ff:fe0a:5bfc	Node2	quorum-manager
2	Node3	2001:192::42f2:e9ff:fe0a:5094	Node3	quorum-gateway

3	Node4	2001:192::42f2:e9ff:fe0a:619c	Node4	quorum-manager-gateway
4	Node5	2001:192::42f2:e9ff:fe0a:5794	Node5	quorum

Note: Here, the AFM cache cluster and AFM remote or home cluster is converted to IPv6.

Existing connection over IPv4

1. `mmafmctl fs1 getstate -j ip1`

Fileset Name numExec	Fileset Target	Cache State	Gateway Node	Queue Length	Queue
-----	-----	-----	-----	-----	
ip1 57588	nfs://192.168.38.101/fileset	Active	Node3	0	

2. Performing failover by using `mmafmctl failover --target-only` option.

```
mmafmctl fs1 failover -j ip1 --new-target [2001:192::210:18ff:fec6:ecd8]:/fileset --target-only
mmafmctl: Performing failover to nfs://[2001:192::210:18ff:fec6:ecd8]:/fileset
Fileset ip1 changed.
```

Note: `--target-only` option changes the Ips of the target. When the `--target-only` option is not provided, it synchronizes all the data from Cache to target.

3. Create a file in a fileset and make sure that the relation becomes active.

```
echo filedata > /gpfs/fs1/ip1/file1
```

```
mmafmctl fs1 getstate -j ip1
```

Fileset Name numExec	Fileset Target	Cache State	Gateway Node	Queue Length	Queue
-----	-----	-----	-----	-----	
ip1	nfs://[2001:192::210:18ff:fec6:ecd8]/fileset	Active	Node3	0	3

Creation of mapping for parallel data transfer by using IPV6 address and hostname

To enable parallel data transfer on IPV6, both home clusters nodes and cache cluster nodes must be running on IPV6. A Gateway Node that runs IPV6 must be able to communicate to the individual home export node to synchronize data across the clusters.

Mapping of home cluster nodes and gateway nodes at cache that are running on IPV6 can be mapped together either by using IPV6 address or the hostname.

Examples of creating mapping by using IPV6 address and hostname

1. **Mapping creation by using IPV6 address**

As you start creating mapping by using IPV6 addresses, IPV6 address must be written inside a square bracket '['].

The following example demonstrates how to create mapping by using IPV6 address:

```
mmafmconfig add map1 --
export-map [2001:192::42f2:e9ff:feaf:6aa8]/[2001:192::42f2:e9ff:feaf:85a0],
[2001:192::42f2:e9ff:feaf:6c30]/[2001:192::d756:f6d2:2710:330f]
```

System output:

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

2. **Mapping creation by using hostname**

Creation of mapping by using hostname is exactly similar to IPv4 mapping creation.

The following example demonstrates how to create mapping by using the hostname:

```
mmafmconfig add mapx1 --export-map c80f3m5n06/c80f2m5n02,c80f3m5n05/c80f5m5n14
```

System output:

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

In the preceding command example, c80f3m5n06 and c80f3m5n05 are export nodes at the home cluster and c80f2m5n02 and c80f5m5n14 are Gateway nodes at the cache cluster.

In the preceding examples, all the hostname resolution is pointing to the IPv6 address.

Note:

- Mix of IPv6 and IPv4 mapping is not supported.
- For more information about Parallel data transfer, see **Parallel data transfers**.

After the mapping is created, user can create an AFM fileset by using mapping. The following command demonstrates how a user can create an AFM fileset by using mapping:

```
mmcrfileset <FS> <fset_name> -p afmMode=<AFM Mode>,afmTarget=<protocol>://<Mapping>/<remoteFS_Path>/<Target> --inode-space new
```

AFM limitations

Active File Management (AFM) limitations are as follows:

- If you change an AFM SW/IW home export from Kernel NFS or CNFS to the new IBM Spectrum Scale CES in 4.1.1 or later, AFM does not recognize the home. The best way to manage this situation is to run the **mmafmctl failover** command. See the IBM Spectrum Scale documentation for more details.
- AFM RO/LU home export cannot be changed from Kernel NFS or CNFS to new IBM Spectrum Scale CES in 4.1.1. or later.
- Customers who enabled DMAPI/IBM Spectrum Protect for Space Management at the AFM home cluster must be running at RHEL 6.3, or later, or SLES 11 SP2, or later, on the cache cluster. There is a bug that is identified in earlier levels of RHEL, which requires that the exported path at home be excluded from DMAPI/IBM Spectrum Protect for Space Management. For more information, see [IBM Support](#).
- Code upgrade limitations:
 - If there are AFM filesets that use the Kernel NFS backend to communicate with a GPFS home that is running V4.1 or earlier, upgrading home to IBM Spectrum Scale V4.1.1 or later causes the AFM filesets to disable synchronization to home with a message as follows:

```
GPFS: 6027-3218 Change in home export detected. Synchronization with home
is suspended until the problem is resolved
```

Note: This upgrade affects only AFM filesets that were originally created with a GPFS home that runs V4.1 or earlier. This upgrade does not affect AFM filesets that were created with a home that runs IBM Spectrum Scale V4.1.1 or later. This issue is fixed in 4.1.1.4 and 4.2.0.1 available at Fix Central ([IBM Fix central](#)).

- AFM filesets do not support the gpfs API – **gpfs_prealloc**.
- Due to a known issue with AFM SMB integration, the cache cluster might experience queued messages on gateway nodes. As a result, SMB clients might experience a performance impact on data transfers.
- The parallel data transfer by using multiple remote mounts feature has a known issue with newly released Linux distributions. If you want to upgrade cluster nodes, disable this feature on RHEL 8.1 (kernel 4.18) and Ubuntu 18.04.4 (kernel 5.3) levels and later.

AFM and AFM DR limitations

The common limitations for both AFM and AFM DR functions include:

- The following table shows the OS and architecture support matrix and supported AFM function:

Table 8. Supported function and supported OS/architecture			
OS and architecture	Fileset	Node roles	Comments
Linux on Power® and x86	SW/IW/RO/LU primary/secondary	Gateway Application Manager	
Linux on Z	SW/IW/RO/LU primary/secondary	Gateway Application Manager	
AIX on Power	SW/IW/RO/LU primary	Gateway Application	AIX cannot be an AFM home site or an AFM-DR secondary site.

- An AFM home site that is not an IBM Spectrum Scale cluster:
 - An AFM cache site can be configured to connect with any home site that provides access to the home data via the NFSv3 protocol. The home site is not required to be an IBM Spectrum Scale cluster.
 - In this scenario, the **mmafmconfig** command cannot not be issued on the AFM home site. Therefore, the AFM cache site cannot cache or update extended attributes (EAs), access control lists (ACLs), file sparseness, or AFM psnaps.
 - The IBM Spectrum Scale support team will help customers to investigate problems in this scenario but only to address issues with the AFM cache site that runs IBM Spectrum Scale. Any problems at the AFM home site, or its file system, must be addressed by the customer.
 - For AFM-DR configurations, both the primary and secondary sites must be IBM Spectrum Scale clusters.
- An AFM home site that runs IBM Spectrum Scale on AIX:
 - The **mmafmconfig** command is not supported on AIX. Therefore, the AFM cache site cannot cache or update extended attributes (EAs), access control lists (ACLs), file sparseness, or AFM psnaps.
 - AFM does not support AIX nodes as an exporter of NFS. That is, AFM does not support NFS mounting exports that are reached via a system that runs on AIX.
 - Because of the previous limitation, AIX nodes cannot act as an AFM home site or as an AFM-DR secondary site.
 - AIX is not supported as a gateway node for AFM fileset modes.
 - AIX is not supported as an AFM home site or AFM-DR secondary site.
- The **mmclone** command is not supported on AFM cache and AFM DR primary filesets. Clones that are created at home for AFM filesets are treated as separate files in the cache.
- Quality of service is not tested on AFM and AFM Async DR filesets.
- Connecting an AFM-DR fileset to IBM Spectrum Protect for Space Management use case is not supported.
- The NSD protocol for communication does not support AFM and AFM DR filesets on Linux on Z.
- Connecting AFM DR fileset to IBM Spectrum Archive is not supported.
- AFM and Async DR is not supported on clusters that have Windows nodes.
- Cascading relationships with AFM caches and AFM primary filesets are not tested.
- Fileset snapshot restore is not supported for AFM and AFM DR filesets.

- Files in an AFM or AFM DR fileset, which are shared by using SMB, an error message E_ACCESS might appear if SMB holds locks on the files. When the locks are released by SMB, AFM replicates again and processes the files from the AFM or AFM DR fileset.
- The `dm_write_invis()` function is not supported on AFM and AFM DR filesets.
- The `dm_read_invis()` function is not supported on AFM and AFM DR filesets, if a file is not cached.
- The File Audit logging feature is not supported on AFM or AFM DR filesets.
- **Immutability** and **appendOnly** features are not supported on AFM filesets.
- AFM and AFM DR filesets do not replicate security, system, or trusted extended attributes.
- The **--iam-mode** option is not supported on AFM filesets.
- AFM and AFM DR do not support DM Punch Hole method (GPFS API '**dm_punch_hole**').
- To enable the Kerberos security for an AFM-DR fileset, the AFM-DR secondary must be given the read/write access.
- If a file is evicted from an AFM fileset, the snapshot file of the evicted file contains zeros.
- Eviction is not supported on an AFM-DR primary fileset and an AFM-DR secondary fileset.
- The following file attributes are maintained locally in the cache but these are not replicated:
 - Control attributes
 - Direct I/O
 - Replication factors
 - Fileset quotas
 - Storage pool flags
 - Special file types such as FIFO, socket, block, character device
- Hard links can be created on the cache. Hard links on the home are displayed as individual files in the cache during the lookup. To prefetch hard links from home, run the **--metadata-only** option as the first operation.
- File locking across the cache and the home is not supported.
- User extended attributes, ACLs, and sparse files supported only on the home where the **mmafmconfig** command is run.
- Parallel data transfer is supported only in Linux-only clusters. If a home is a mix of architectures (x86 and ppc), parallel data transfer works only for the set of nodes that belong to any one architecture, depending on which architecture serves the data transfer first.
- 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 Spectrum 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.

AFM-based Asynchronous Disaster Recovery (AFM DR)

The following topics introduce you to AFM-based Asynchronous Disaster Recovery (AFM DR).

The following figure illustrates the AFM disaster recovery process.

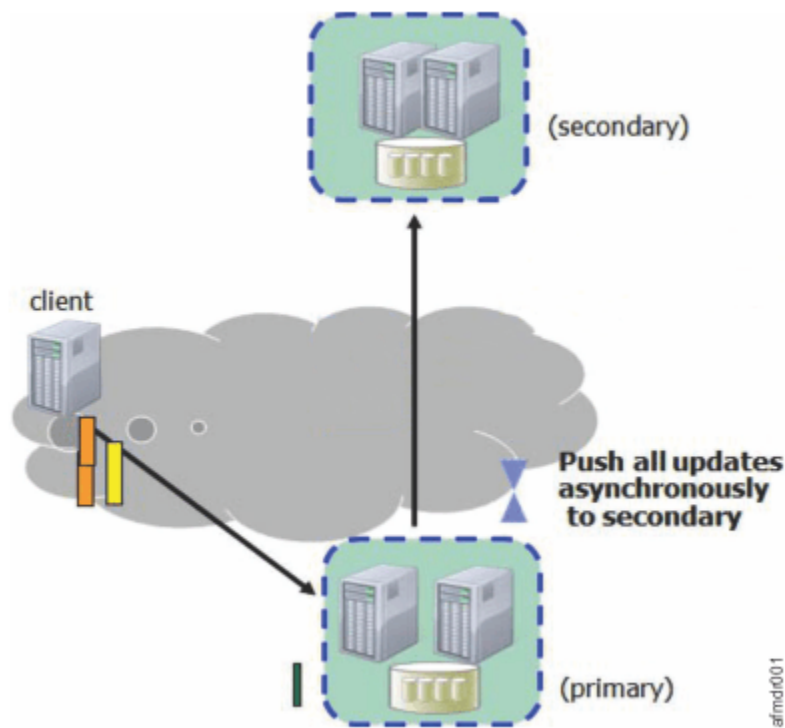


Figure 15. Asynchronous disaster recovery

Introduction

Active File Management-based asynchronous disaster recovery (AFM DR) is a fileset-level replication disaster-recovery capability.

Important: The 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 feedback, IBM decided to disable the AFM DR by default. You should contact IBM Spectrum Scale support at scale@us.ibm.com to have your use case reviewed. IBM helps to optimize your tuning parameters and enable the feature. Please include this message when you contact to IBM support.

For more information, see [Flash \(Alert\): IBM Spectrum Scale \(GPFS\) V4.2 and V4.1.1 AFM Async DR requirement for planning](#).

The disaster recovery solution includes:

- Providing business stability during a disaster
- Restoring business stability after the disaster is repaired.
- Enduring multiple disasters
- Minimizing data loss because of a disaster

AFM-based asynchronous disaster recovery is an AFM-based fileset-level replication disaster-recovery capability that augments the overall business recovery solution. This capability is a one-to-one active-passive model and is represented by two sites: primary and secondary.

The primary site is a read/write fileset where the applications are currently running and has read/write access to the data. The secondary site is read-only. All the data from the primary site is asynchronously synchronized with the secondary site. The primary and secondary sites can be independently created in storage and network configuration. After the sites are created, you can establish a relationship between the two filesets. The primary site is available for the applications even when communication or secondary fails. When the connection with the secondary site is restored, the primary site detects the restored connection and asynchronously updates the secondary site.

The following data is replicated from the primary site to the secondary site:

- File-user data
- Metadata including the user-extended attributes except the inode number and a time
- Hard links
- Renames

The following file system and fileset-related attributes from the primary site are not replicated to the secondary:

- User, group, and fileset quotas
- Replication factors
- Dependent filesets

AFM DR can be enabled on GPFS-independent filesets only.

Note: An independent fileset that has dependent filesets cannot be converted into an AFM DR fileset.

A consistent view of the data in the primary fileset can be propagated to the secondary fileset by using fileset-based snapshots (psnaps). Recovery Point Objective (RPO) defines the frequency of these snapshots and can send alerts through events when it is unable to achieve the set RPO. RPO is disabled by default. The minimum time that you can set as RPO is 720 minutes. AFM-based asynchronous DR can reconfigure the old primary site or establish a new primary site and synchronize it with the current primary site.

Individual files in the AFM DR filesets can be compressed. Compressing files saves disk space. For more information, see *File compression* in the *IBM Spectrum Scale: Administration Guide*.

Snapshot data migration is also supported. For more information, see *ILM for snapshots* in the *IBM Spectrum Scale: Administration Guide*.

When a disaster occurs on the primary site, the secondary site can be failed over to become the primary site. When required, the filesets of the secondary site can be restored to the state of the last consistent RPO snapshot. Applications can be moved or failed over to the acting primary site. This application movement helps to ensure stability with minimal downtime and minimal data loss. This makes it possible for applications to eventually be failed back to the primary site as soon as the (new) primary is on the same level as the acting primary.

Note: AFM DR does not offer any feature to check consistency of files across primary and secondary sites. However, you can use any third-party utility to check that consistency after files are replicated.

You can simultaneously configure a site for continuous replication of IBM Spectrum Scale data along with AFM DR site. With IBM Spectrum Scale continuous replication, you can achieve a near disaster recovery and a far disaster recovery with AFM DR site.

Recovery time objective (RTO)

Recovery time objective (RTO) is the time that is taken to convert a secondary fileset into an active primary fileset and move applications to that site after a disaster.

After a disaster at the primary site, the secondary site becomes the acting primary by running the **mmafmctl failoverToSecondary** command. The `--norestore` option is set by default. The secondary fileset is converted to acting primary but avoids restoration of data from the last RPO snapshot and maintains the existing data. As an administrator, you can run the **mmafmctl failoverToSecondary** command with the `--restore` option to clean the existing data at the secondary fileset and restore all data from the last RPO snapshot.

RTO can be estimated if the administrator knows the volume of data that is generated in an RPO interval. The volume of data that must be restored from the last snapshot depends on the number of files that changed in an RPO interval. Because a snapshot restore requires a complete scan of the snapshot to determine which files changed. The failover time is one of the components that an administrator must use to calculate the RTO for the application.

Modes and concepts

AFM DR uses the same underlying infrastructure as AFM. AFM DR is characterized by two modes: the fileset in the primary cluster uses the primary mode and the fileset in the secondary cluster uses the secondary mode.

AFM DR is supported over both NFS v3 and GPFS protocol. The primary fileset is owned by the primary gateway, which communicates with the NFS server on the secondary side. The primary-secondary relationship is strictly one-to-one.

AFM revalidation does not apply to primary filesets. All files are always cached because primary is the only writer and secondary is in the read-only mode.

You can convert the SW/IW relationship to a DR relationship. However, you cannot convert a DR relationship to an SW/IW relationship.

AFM-based Asynchronous Disaster Recovery features

The following sections describe the AFM-DR features.

The following AFM features are offered on AFM DR filesets:

- Force flushing contents before Async Delay
- Parallel data transfers
- Peer snapshot - psnap
- Gateway node failure and recovery
- Operation with disconnected secondary
- Using IBM Spectrum Protect for Space Management
- Disabling AFM DR
- Using AFM DR with encryption
- Stop and start replication on a fileset

You can use **mmbackup** command to back up all files from primary, as all files are in a cached state on the primary fileset. Similar to AFM filesets, IBM Spectrum Protect can be connected to primary or secondary, or both sides. When IBM Spectrum Protect is connected to the primary side, set `AFMSKIPUNCACHEDFILES yes` in `dsm.sys` file. AFM features such as revalidation, eviction, prefetch, partial file caching, expiration, resynchronization, failover, and showing home snapshots are not offered on AFM DR filesets.

Immutability and appendOnly for AFM-DR filesets

By enabling an integrated archive manager (IAM) mode on an Active File Management Disaster Recovery (AFM-DR) primary and secondary fileset, you can set the Immutability and appendOnly flags on a file or a directory in an AFM-DR fileset.

You cannot change or rename an immutable file. On a file or directory with the appendOnly flag enabled, you can do append operations, but you cannot do delete, modify, or rename operations.

After you set these flags on a file or a directory in an AFM-DR primary fileset, AFM automatically synchronizes these metadata flags on the secondary fileset. You can set the Immutability and appendOnly flags independently. If both Immutability and appendOnly flags are set on a file, the Immutability flag takes effect.

To enable the Immutability and appendOnly flags, you need to enable an IAM mode on an AFM-DR primary and secondary fileset. All IAM modes are supported. The behavior of the Immutability and appendOnly flags is the same for a GPFS file and a GPFS directory. To configure an IAM mode, see *Enabling integrated archive manager (IAM) modes on AFM-DR filesets* in the *IBM Spectrum Scale: Administration Guide*.

For more information about these flags, see *Immutability and appendOnly features* in the *IBM Spectrum Scale: Administration Guide*.

Note: You must do the following upgrade for the Immutability and appendOnly flags:

- Upgrade all nodes in a cluster to the latest release level.
- Upgrade the file system version to the latest version.
- Upgrade the cluster minRelease version to the latest version.

If an AFM-DR primary fileset and an AFM-DR secondary fileset has a linked dependent fileset, ensure that the same IAM mode, which is set on the primary fileset and the secondary fileset, is set on all linked dependent filesets.

If an AFM-DR primary and an AFM-DR secondary fileset has one or more linked dependent filesets and you want to set the compliant mode, do the following steps:

1. Set the compliant IAM mode on the primary fileset and all linked dependent filesets.
2. Set the compliant IAM mode on the secondary fileset and all linked dependent filesets.

Note: To enable IAM mode on a dependent fileset, you need not to run the **mmafmctl stop** command or the **mmafmctl start** command. These commands are needed only to set an IAM mode on an AFM-DR primary fileset.

RPO snapshots

Recovery point objective (RPO) snapshots are peer snapshots that are taken at the same time on the primary and the secondary sides. RPO is disabled by default, and the minimum value you can set is 60 minutes and increment in multiples of 5 minutes. You can update the **afmRPO** parameter while you create the primary to change the interval, or after you created the fileset by using the **mmchfileset -p** command with the **afmRPO** parameter.

An example of the command -

```
mmchfileset <FileSystemName> <filesetname> -p afmRPO=<time_in_minutes>
```

The appropriate RPO intervals for each application setup are determined by the following factors:

- The rate of data change
- The fileset parameters such as **afmAsyncDelay** and **afmNumFlushThreads**
- The network bandwidth between the two sites

Each RPO interval triggers a snapshot at fileset level on both the primary and the secondary sides, that results in the file system being quiesced. Quiescing the file system is a data-transfer intensive action. Therefore, as a performance optimization, if multiple caches have similar RPOs, their snapshots are batched together so that the file system is quiesced only once. As further optimization, RPO snapshots are only taken for primary filesets whose data or metadata is modified since the last RPO period.

The **afmAsyncDelay** parameter specifies the minimum time that the primary site must wait before flushing the pending data that needs to be transferred to the secondary site. An RPO request in queue flushes the queue before you create a snapshot on the secondary site. The general guidelines for the **afmAsyncDelay** parameter is to set the asynchronous delay less than the RPO interval.

The **afmRPOMiss** event occurs when RPO is missed because of a network delay or failure to create the RPO snapshot on the secondary site. If RPOs are missed, AFM waits for the expiration of the next RPO and during next RPO, a message is logged in **mmfs.log** and a new RPO snapshot is created. Failed RPOs are queued on the gateway again, and are run again at the secondary site until they succeed. At any point in time, two RPO snapshots that are not **psnap0** are retained on both sides. If RPOs are not played at the secondary due to some reason and primary does not get acknowledgment for the create from the secondary, the RPO is also deleted from primary. To improve the performance of more than one fileset taking RPOs at the same time, the RPO snapshots are batched together. In this process, the RPOs of few filesets might get slightly adjusted to use the batching facility.

Normal RPO deletions from the primary site are performed and queued when the new RPO snapshots are received. While there is every attempt to ensure that the delete operations are successful on the secondary site, there can be some extra snapshots as some delete operations might not be successful. In such cases, you must manually delete the extra snapshots on the secondary site by using **mmdeletesnapshot -p**. Apart from automatic RPO snapshots, you can create user snapshots by running the **mmpsnap create** command with the **-rpo** option. These snapshots are intermediate between scheduled RPOs and are not deleted during the RPO snapshot delete process.

When the primary site fails, you can roll back the last consistent snapshot present on the secondary site while converting it to the current primary site. However, it is recommended to convert the secondary as-is into the acting primary because in most cases the data in the secondary is always greater than or equal to the last RPO snapshot. When applications are moved to the current primary fileset after the failover operation, RPO snapshots are not taken because no secondary is associated with the current primary. Therefore, RPO is not applicable to the current primary site. The gateway node mapped to serve a fileset is called the primary gateway of the fileset. The primary gateway acts as the owner of the fileset. If the primary gateway fails, another gateway node takes over the fileset as primary gateway. The recovery is triggered on the new primary gateway while taking over the fileset. If the next RPO snapshot is due at the time when recovery gets triggered, a new RPO snapshot is taken and used for recovery. Whenever RPO snapshots are used for recovery, the normal cleanup process of the older RPO snapshots does not take place. Therefore, some extra RPO snapshots are temporarily displayed on the primary and secondary sites and are cleaned up on subsequent RPO intervals. No regular RPO snapshots are taken on a fileset that is running recovery until the recovery process is complete.

All user-created and system-created RPO snapshots except the **psnap0** belonging to an active primary-secondary must be deleted before the primary-secondary is deleted. You can change the RPO interval by running **mmchfileset**. The time for the next RPO snapshot (**Tnext**) is calculated by adding the RPO interval that you set (**Irpo**) to the time at which the previous snapshot occurred (**Tprev**): **Tnext = Tprev + Irpo**. If no file in the RPO snapshot changes during the interval, AFM does not take the snapshot, and the time for the next RPO snapshot is calculated as **Tnext = Tnext + Irpo**.

Note: The RPO interval is not added to the time at which you set the RPO interval, but to the time of the last RPO snapshot. For example, the previous snapshot was at 9:00 AM and the RPO interval is 720 minutes. If you change the RPO interval to 780 minutes during the 12 hours from 9:00 AM, the next RPO snapshot occurs after 780 minutes from 9:00 AM.

The RPO management deletes the oldest snapshot when it creates a new snapshot but it never deletes the **psnap0**. The **mmpsnap** command can be run to delete **psnap0**. By deleting **psnap0**, storage utilization improves because data blocks used by **psnap0** are not held down. These data blocks can be significant over a period of time. Also, deletion of **psnap0** can improve the performance of subsequent creation and deletion of RPO snapshots. However, you must delete **psnap0** only when other RPO snapshots are present in the primary and secondary filesets to handle disaster recovery. If **psnap0** is deleted without any other RPO snapshots, data is lost. Also, **psnap0** can be deleted from secondary in cases like failing over to the secondary, and a new primary being set up.

How to decide if you need RPO

Failover

RPO should be enabled if you need to go back in time after a failover by restoring the data on the secondary from the last RPO snapshot. But this is NOT recommended, as the data in the secondary is always greater than or equal to last RPO snapshot. Hence in most cases, RPO is not needed for failover.

Failback

If the primary comes back after a temporary failure, you would only need to copy the data that has changed in the secondary to the primary. For this, RPO function needs to be enabled. However, the RPO can be a large value, such as 24 hours. This value will dictate how much data is copied back to primary from the secondary during a failback. The higher the value, the larger the amount of data copied.

Factors for deciding the RPO interval

RPO interval dictates how often snapshots are created on the primary and secondary. The entire file system must be quiesced before taking a snapshot, even if the snapshot is for a single fileset in a massive file system.



CAUTION: Setting RPO intervals on primary fileset has significant implications on the performance of the file system.

- Classify your filesets based on how critical the data is - the more critical the data the lower the async delay.
- To calculate this approximately, review the frequency/pattern of data generation at the primary and considering the number of GW nodes, network latency and other parameters, predict the time taken for this data to get synchronized with the secondary. The RPO must be set to this time.
- Do not set many filesets to the same RPO interval.

Role reversal

When a planned or unplanned failover occurs, the role reversal process reverses a secondary site to an acting primary site and the old primary site to the secondary site. The role reversal is recommended to handle the failover.

When a primary site fails, applications can move to a secondary site after the secondary site is promoted to the primary role. The secondary site acts as a primary site for the application. After reversal of role, the secondary site can continue to act as a primary site until the old primary site is back. When the failed site (old primary site) is restored, it takes over the role of the secondary site.

In the traditional “Failback to primary” method, when the failed primary site is restored, workloads are transferred from the acting primary site (old secondary site) to the restored primary site. This method is recommended when the rate of change at the primary is not high, and workloads can be easily transferred between two sites.

In the role reversal method, the secondary site permanently acts as a primary site. After the primary site is restored, it is designated as a secondary site. The role reversal method is recommended when the rate of change at the primary is high, and the filesets are huge. If you adopt the role reversal method, ensure that you remove extra files and old snapshots from the new secondary site.

Important: For a planned or unplanned failure, it is recommended to use the role reversal method instead of failover and failback methods. The failover and failback methods will be deprecated soon.

To reverse the role, do the following steps:

1. When you plan the role reversal, ensure that the primary and secondary sites are in synchronization.
2. Issue the following command on the secondary site after the primary site failure:

```
# mmadmctl Device failoverToSecondary -j FilesetName [--norestore |--restore ]
```

3. After the secondary site is promoted to the primary site, move applications on the primary site.
4. After the old primary site is restored, prepare the old primary site to promote to the secondary site.
5. If the role reversal is unplanned, ensure that whether the primary site has dirty files. These files were created or modified but were not replicated to the secondary site because of failures.
 - Do not delete the dirty files that were available at both sites but were not in sync because of the primary site failure. After the role reversal, the acting primary site overwrites these files on the secondary site (the old primary site) when the site is restored and configured.
 - Dirty files that were newly created at the old primary site but not replicated to the secondary site because of the primary site failure. These files are extra files at the old primary (secondary) after the role reversal.
 - These files are overwritten if the same named files are created at the acting primary site and replicated to the secondary (old primary) site.

- If you choose again the old primary site as a primary site, then these created files are replicated to the secondary site.
6. To prepare the old primary site, do the following steps:
 - a. Unlink the fileset by issuing the following command:

```
# mmunlinkfileset fs fileset -f
```

- b. Disable the role of fileset at the primary site by running the following command:

```
# mmchfileset fs fileset -p afmTarget=disable
```

- c. Link back the fileset by issuing the following command:

```
# mmlinkfileset fs fileset -J path_to_fs/fileset
```

- d. In case of an unplanned role reversal, some changes are not yet replicated from an old primary to a new primary site. However, before the replication, the primary that have an active queue fails. In this case, run the **failoverToSecondary** parameter as in step “2” on page 102. If you want to keep the old primary, which is the reversed secondary site, to the same RPO snapshot level, run the **mmrestorefs** command to revert the fileset to the RPO snapshot level for the application consistency.

```
# mmrestorefs fs latestRPO -j fileset
```

Note: If the old primary is not restored, it does not cause the data inconsistency at a file system level or at a fileset level. But, it might cause data inconsistency for the application. For example, a file on an old primary with new data changes that are not replicated to an old secondary. If before this replication completes, there is a failure at the old primary and this data does not replicate to the old secondary, and the old primary is not reverted to the RPO snapshot level during the role reversal, the both sites are synchronized because of the following scenarios:

- Some data of the file is replicated to the old secondary, but some data is still being replicated. In this case, the reverse relationship establishment overwrites the file from the reversed primary to the reversed secondary, and both the sites are synchronized.
- The file does not exist on the old secondary. It is still in a queue on the old primary when it failed. In this case, one of the following things might happen:
 - In the reverse relationship, the reversed primary has a file with the same name. Because of the same name the new file from the reversed primary is overwritten to the reverse secondary, and both the sites are synchronized.
 - The reverse relationship does not affect the new file. Therefore, the reversed primary does not know whether the new file is on the reversed secondary. For such a new file, when the role is reversed second time, the original primary becomes primary again. The relationship establishment synchronizes the new file to the original secondary and both filesets data becomes consistent at the fileset level and on a snapshot.

7. Get the primary ID of the fileset at the acting primary site by issuing the following command:

```
# mmafmctl fs getPrimaryId -j fileset
```

8. Convert the fileset of the old primary site to the fileset of the secondary site by issuing the following command at the old primary:

```
# mmafmctl fs convertToSecondary -j fileset --primaryid PrimaryID
```

9. Prepare to export this secondary fileset path for use by the acting primary fileset.

10. Issue the following command at the acting primary to build the relationship:

```
# mmafmctl fs changeSecondary -j fileset --new-target oldPrimary:filesetpath --inband
```

11. Remove old snapshots from the old primary because the old snapshots contain old data.

The primary site (the old secondary site) is now the primary site and the old primary site is the secondary site. The roles of the primary site and the secondary site are interchanged and the relationship is restored.

Note: The role reversal method is used if filesets are large, and a high rate of change at the primary site.

Failover to the secondary site

When the primary site stops functioning, all applications must move to the secondary site to ensure continuity. Run the following command on the secondary to convert the secondary site into acting primary site: `mmafctl <FileSystemName> failoverToSecondary -j <FilesetName> [--norestore | --restore]`.

You can use the `--restore` option to restore the data from latest snapshot or the `--norestore` option to keep the all existing data during the failover process on the secondary site. The `--norestore` option is the default option. After the failover, the secondary site acts as a primary site and hosts applications.

The customer must ensure that the NFS export used for the secondary site is not removed so that after failback to the primary, the original primary site can communicate with the secondary site, which is the current primary. Also, the RPO snapshots existing on the current primary must not be deleted because the RPO snapshots are used during the failback process when the original primary site starts functioning again.

RPO snapshots are temporarily disabled on the active primary site. The new data is not replicated to any other site until a new primary is set up, or the original primary is reinstalled, and the current primary is converted back to the secondary site.

Failing back to the old primary site

The old primary site must be restored when it is repaired.

Important: For a planned or unplanned failure, it is recommended to use the role reversal method instead of failover and failback methods. For more information about the role reversal, see [“Role reversal” on page 102](#).

Complete the following steps to restore the old primary site when it is repaired and it is back online after the disaster:

1. Issue the following command on the old primary:

```
# mmafctl Device failbackToPrimary -j FilesetName { --start | --stop }[--force]
```

The `--start` option restores the primary to the contents from the last RPO on the primary before the disaster. With the `--start` option, the primary is in the read-only mode. This mode avoids accidental corruption until the failback process is completed. After the old primary site starts functioning again, all RPOs that were present before the disaster can be accessed. If a common RPO snapshot `psnap0` is not present, the old primary site can be converted to a normal GPFS fileset. To set up the primary site, see the steps in [“Failing back to the new primary site” on page 105](#).

If the `--start` option that is run on the fileset is unsuccessful, next the `--start failbackToPrimary` option might not be allowed. You can use the `--force` option to start failback again.

While the primary is coming back, as an administrator ensures that I/O does not occur on the primary fileset before the start of the `failback --start` process.

2. Issue the following command on the old primary:

```
# mmafctl Device { applyUpdates | getPrimaryId } -j FilesetName
```

This command applies differences that are created by applications on the old primary site as the current primary site took over the applications.

All the differences can be brought over in a single or multiple iterations. For minimizing the application downtime, this command can be run repeatedly to synchronize the contents of the original primary site

with the current primary site. When the contents on both the sites are as close as possible or have minimal differences, applications must take a downtime and this command must be run one last time. applyUpdates might fail with an error during instances when the acting primary is overloaded. In such cases, the command needs to be run again.

3. On the old primary, complete the failback process by running **mmafmctl** with the **failbackToPrimary --stop** option.

With this command, the fileset is in the read/write mode. The primary site is ready for starting the applications. If the **--stop** option of the failback does not complete due to errors and you cannot stop the failback, it can be forced to stop with the **--force** option.

4. Convert the current primary site back to the secondary site, and set the primary ID. Unlink the acting primary site, change it to secondary by issuing the following command on the acting primary (secondary):

```
# mmchfileset device fileset -p afmMode=secondary -p afmPrimaryID=primaryid)
```

NFS can be restarted on the secondary site to ensure that the secondary export is accessible to the primary site. The primary and secondary sites are connected back as before the primary disaster and all data from the primary is played on the secondary site. Regular RPO also resumes on the primary site.

Failing back to the new primary site

You can configure a new primary if the old primary site stopped functioning.

Important: For a planned or unplanned failure, it is recommended to use the role reversal method instead of failover and failback methods. For more information about the role reversal, see [“Role reversal” on page 102](#).

Complete the following steps to configure a new primary site:

1. Create a GPFS fileset on the new primary site. This fileset is configured as the new primary.
2. Create the latest snapshot from the acting primary by running the **mmafmctl** command with the **replacePrimary** option.
A new psnap0 with all the latest contents is created on the current primary server. This snapshot is used in the subsequent steps in this task.
3. Copy the contents from the snapshot to the new primary site by using **scp** or other means outside of AFM.
4. The snapshot that is created from the acting primary is an argument to the **--secondary-snapname** option. Issue the following command to convert the fileset on the primary site:

```
# mmafmctl Device convertToPrimary -j FilesetName  
[--afmtarget Target { --inband | --secondary-snapname SnapshotName}]  
[ --check-metadata | --nocheck-metadata ][--rpo RPO] [-s LocalWorkDirectory]
```

The primary ID is generated and a psnap0 with the same name is created on the primary site. After the ID generation and the snapshot creation, complete all the steps in [“Failing back to the old primary site” on page 104](#).

Changing the secondary site

This topic lists the steps to follow when the secondary site stops functioning.

Complete the following steps if the secondary site stops functioning:

1. Create a GPFS independent fileset on the new secondary site.
2. Copy the data on the primary site to the new GPFS fileset by using **ftp** or **scp**. This copying data is known as outband trucking.
3. Run the following command to establish a new secondary site for the primary site:

```
mmafmctlDevice changeSecondary-j FilesetName
--new-target NewAfmTarget [ --target-only |--inband ]
[-s LocalWorkDirectory]
```

The **--inband** option is used depending on the method that is being used to truck data. If you want to change the mount path or the IP address in the target path, you can use the **--target-only** option. The new NFS server must be in the same home cluster and must be of the same architecture as the existing NFS server in the target path. The **--target-only** option must not be used if the home is changed completely, or if you want to change the protocol while you are using the same home cluster. While you are using the same secondary path, if you want to change only the target protocol from GPFS to NFS and vice versa, you can use **mmafmctl changeSecondary** command.

4. Convert the GPFS fileset on the new secondary site to a secondary and set the primary ID by running the **mmchfileset** command or the **mmafmctl** command with **convertToSecondary** option.

Ensure that the NFS export on the secondary site is accessible on all the gateway nodes on the primary cluster if NFS is used for defining AFM target on the primary site. If the GPFS protocol is used for the target, the secondary file system must be mounted on all gateway nodes on the primary site.

After the primary and secondary sites are linked, the `psnap0` queued from the primary fileset is played on the secondary fileset. The new secondary site is now linked to this primary site. Ensure that the primary file system is mounted on all gateway nodes and new secondary file system is mounted on all the nodes in the secondary cluster.

Using the primary fileset to make changes to the secondary fileset

1. Replace the secondary with a new empty secondary where the new target uses NFS/NSD/mapping.

The administrator must run the **mmafmctl changeSecondary** with **--inband** option to point to the new secondary. The new secondary is expected to be empty. If the new target is a mapping, **changeSecondary** does not split I/Os and queues them as normal requests without parallel data transfer.

2. Replace any of the following on an existing secondary:

- Replace the communication protocol (NSD/NFS):

The administrator must run the **mmafmctl changeSecondary** with **--inband** option by using the new target protocol.

Note: Only the protocol changes and not the secondary path.

- Enable or disable parallel data transfer PIO by shifting between NFS/NSD and a mapping:

The administrator must run the **mmafmctl changeSecondary** with **--inband** option, by using the new target protocol or mapping.

Note: Only the protocol changes and not the secondary path.

- Replace only the IP address or NFS server by using the same communication protocol and secondary path:

The administrator must run the **mmafmctl changeSecondary** command with the **-target-only** option.

Note: Only the IP or NFS server changes. The IP/NFS server must be on the same secondary cluster and must be of the same architecture as the old NFS server.

AFM DR limitations

AFM DR limitations include:

- The 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 feedback, IBM decided to disable the Active File Management-based Asynchronous Disaster Recovery feature (AFM DR) by default. Customers that are deploying the AFM DR feature must first review their

deployments with IBM Spectrum Scale development. You should contact Spectrum Scale Support at scale@us.ibm.com to have your use case reviewed. IBM will help optimize your tuning parameters and enable the feature. Include this message when you contact IBM Support.

Note: This limitation does not apply to base AFM support. This limitation applies only to Async DR available with the IBM Spectrum Scale Advanced Edition V4.1.1 and later.

You must have the following information available when you request for a development led feasibility assessment for Async DR-related proposals.

- Detailed use case description including any application description for applications that are running in DR filesets.
- Network latency and bandwidth between sites.
- Scale of the solution that includes the number of filesets, number of files, type of files (small or large), any special file usage (clones or encryption).
- RPO time.
- Rate of change of data that needs to be transferred between two RPOs from Async DR primary site to secondary.
- Number of Async DR filesets.
- Any plans to use snapshots in Async DR filesets.
- If you change an AFM DR Secondary export from Kernel NFS or CNFS to the new IBM Spectrum Scale CES in 4.1.1 or later, AFM does not recognize the secondary. The best way to manage this situation is to run the **mmafmctl changeSecondary** command. See the IBM Spectrum Scale documentation for more details.
- DR administration by using the **mmafmctl** command is not supported when run from the AIX OS.
- AFM ADR (primary/secondary filesets) is not supported on an FPO enabled system.
- AFM features such as revalidation, eviction, prefetch, show home snapshots, partial file caching, expire and unexpire, mmafmctl resync and failover, and IW failback are not applicable for AFM DR filesets.
- Administrative operations such as fileset conversion to primary, failover, and failback can be run at the same time on multiple filesets. However, it is not recommended because commands might fail with a "busy" error. The commands might fail in cases where the system runs out of the resources that are needed to perform multiple operations at the same time. If this error occurs, rerun the failed commands after the conflicting command completes.

AFM and AFM DR limitations

The common limitations for both AFM and AFM DR functions include:

- The following table shows the OS and architecture support matrix and supported AFM function:

<i>Table 9. Supported function and supported OS/architecture</i>			
OS and architecture	Fileset	Node roles	Comments
Linux on Power and x86	SW/IW/RO/LU primary/secondary	Gateway Application Manager	
Linux on Z	SW/IW/RO/LU primary/secondary	Gateway Application Manager	
AIX on Power	SW/IW/RO/LU primary	Gateway Application	AIX cannot be an AFM home site or an AFM-DR secondary site.

- An AFM home site that is not an IBM Spectrum Scale cluster:
 - An AFM cache site can be configured to connect with any home site that provides access to the home data via the NFSv3 protocol. The home site is not required to be an IBM Spectrum Scale cluster.
 - In this scenario, the **mmafmconfig** command cannot not be issued on the AFM home site. Therefore, the AFM cache site cannot cache or update extended attributes (EAs), access control lists (ACLs), file sparseness, or AFM psnaps.
 - The IBM Spectrum Scale support team will help customers to investigate problems in this scenario but only to address issues with the AFM cache site that runs IBM Spectrum Scale. Any problems at the AFM home site, or its file system, must be addressed by the customer.
 - For AFM-DR configurations, both the primary and secondary sites must be IBM Spectrum Scale clusters.
- An AFM home site that runs IBM Spectrum Scale on AIX:
 - The **mmafmconfig** command is not supported on AIX. Therefore, the AFM cache site cannot cache or update extended attributes (EAs), access control lists (ACLs), file sparseness, or AFM psnaps.
 - AFM does not support AIX nodes as an exporter of NFS. That is, AFM does not support NFS mounting exports that are reached via a system that runs on AIX.
 - Because of the previous limitation, AIX nodes cannot act as an AFM home site or as an AFM-DR secondary site.
 - AIX is not supported as a gateway node for AFM fileset modes.
 - AIX is not supported as an AFM home site or AFM-DR secondary site.
- The **mmclone** command is not supported on AFM cache and AFM DR primary filesets. Clones that are created at home for AFM filesets are treated as separate files in the cache.
- Quality of service is not tested on AFM and AFM Async DR filesets.
- Connecting an AFM-DR fileset to IBM Spectrum Protect for Space Management use case is not supported.
- The NSD protocol for communication does not support AFM and AFM DR filesets on Linux on Z.
- Connecting AFM DR fileset to IBM Spectrum Archive is not supported.
- AFM and Async DR is not supported on clusters that have Windows nodes.
- Cascading relationships with AFM caches and AFM primary filesets are not tested.
- Fileset snapshot restore is not supported for AFM and AFM DR filesets.
- Files in an AFM or AFM DR fileset, which are shared by using SMB, an error message E_ACCESS might appear if SMB holds locks on the files. When the locks are released by SMB, AFM replicates again and processes the files from the AFM or AFM DR fileset.
- The `dm_write_invis()` function is not supported on AFM and AFM DR filesets.
- The `dm_read_invis()` function is not supported on AFM and AFM DR filesets, if a file is not cached.
- The File Audit logging feature is not supported on AFM or AFM DR filesets.
- **Immutability** and **appendOnly** features are not supported on AFM filesets.
- AFM and AFM DR filesets do not replicate security, system, or trusted extended attributes.
- The **--iam-mode** option is not supported on AFM filesets.
- AFM and AFM DR do not support DM Punch Hole method (GPFS API '**dm_punch_hole**').
- To enable the Kerberos security for an AFM-DR fileset, the AFM-DR secondary must be given the read/write access.
- If a file is evicted from an AFM fileset, the snapshot file of the evicted file contains zeros.
- Eviction is not supported on an AFM-DR primary fileset and an AFM-DR secondary fileset.
- The following file attributes are maintained locally in the cache but these are not replicated:
 - Control attributes

- Direct I/O
- Replication factors
- Fileset quotas
- Storage pool flags
- Special file types such as FIFO, socket, block, character device
- Hard links can be created on the cache. Hard links on the home are displayed as individual files in the cache during the lookup. To prefetch hard links from home, run the `--metadata-only` option as the first operation.
- File locking across the cache and the home is not supported.
- User extended attributes, ACLs, and sparse files supported only on the home where the **mmafmconfig** command is run.
- Parallel data transfer is supported only in Linux-only clusters. If a home is a mix of architectures (x86 and ppc), parallel data transfer works only for the set of nodes that belong to any one architecture, depending on which architecture serves the data transfer first.
- 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 Spectrum 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.

AFM DR deployment considerations and best practices

For AFM DR deployment, it is important to know the characteristics of AFM DR including advantages and limitations, its deployment method, and best practices.

AFM DR is a file-level asynchronous disaster recovery solution. Replication of data to a secondary cluster is asynchronous, which means that the data that is created on the primary cluster is copied to the secondary cluster with delay. The amount of delay is determined by the network bandwidth between the sites and some IBM Spectrum Scale tuning parameters.

AFM DR copies only the data blocks that changed on the primary cluster to the secondary cluster, which makes it an efficient replication mechanism.

Since data replication is asynchronous by using AFM DR to recover data after a primary cluster failure might cause some data loss. The amount of data that is lost depends on the amount of data that is in the primary queue when a failure occurs. Therefore, AFM DR is not the right disaster recovery solution if your application cannot tolerate possible data loss. If this loss is not acceptable, customers must implement a synchronous disaster recovery mechanism.

Characteristics of AFM DR

The following sections give a brief description of the various characteristics of AFM DR.

Independent filesets in AFM DR

The primary and secondary fileset clusters are independent filesets.

Features

- The disaster protection of data happens at an independent fileset level.
- A single file system can contain multiple AFM DR relationships, one per fileset.

Advantages

- Fileset-level replication allows a high level of granularity and flexibility to protect data.
- An RPO can be specified for each fileset, which allows to set it based on the criticality of the data.
- AsyncDelay can be set on a fileset level to control rate of replication.

Limitations

- Can be defined only on an independent fileset.
- A single AFM DR relationship cannot be created at the file system level.

One-on-one relationships in AFM DR

AFM DR creates a one-to-one relationship between the primary and the secondary filesets.

Advantages

- RPO intervals can be defined per AFM DR relationship.
- AsyncDelay can be specified per AFM DR relationship.
- Quotas can be specified for the primary and secondary filesets.

Limitations

You can create only two copies of the data.

Active-passive relationships in AFM DR

AFM DR creates an active-passive relationship between the primary and the secondary clusters.

Features

- Only the primary can write to the secondary fileset. The secondary is read-only for all other users.
- You can take a snapshot of the secondary fileset and use the secondary as a source to create backups or additional copies.

Advantages

The secondary fileset is protected, so no other program or application other than the primary can write to the secondary.

Limitations

- All operations are allowed on the primary fileset, but only snapshot operations are allowed on the secondary fileset.
- The secondary can be converted into the primary even when the primary is writing to the secondary. There is no protection against an administrator doing a failover operation on the secondary, even if the DR relationship is healthy.
- The administrator must ensure that the primary cannot accidentally write to secondary after a failover.

NFSv3 versus NSD exports in AFM DR

The primary accesses the secondary fileset by using either NFSv3 or NSD protocol.

Features of NFSv3

NFSv3 is a stateless protocol, which is resilient on low-bandwidth and high-latency networks.

Advantages of NFSv3

It is recommended to use NFSv3 because NFSv3 is more tolerant when deployed on an unstable or high-latency network.

Note: It is recommended to use the NSD protocol only if the NFS protocol does not meet the expected performance requirements.

Limitations of NFSv3

For parallel I/O to work by using NFSv3, create a map between the gateway nodes in the primary to the NFS servers in the secondary. For more information about parallel I/O configuration, see *Configuration parameters for AFM* in the *IBM Spectrum Scale: Administration Guide*.

Features of NSD

The NSD protocol is sensitive to packet drops and network latency.

Advantages of NSD

NSD provides high performance to LAN-based installations.

Limitations of NSD

If the secondary cluster is inaccessible, the NSD mount hangs, which in turn might cause the application to hang on the primary cluster.

Trucking features in AFM DR

AFM DR trucking feature can be used to synchronize data between the primary and secondary filesets when the DR relationship is established for the first time.

Features

- AFM DR requires the same data between the primary and secondary before a DR relationship is established by using the peer-to-peer snapshot (PSNAP) mechanism. If the data on the primary and secondary are the same, the peer-to-peer snapshot mechanism creates a special snapshot called PSNAP0. The PSNAP0 is created on both the primary fileset and the secondary fileset.
- Two trucking methods are supported by AFM DR:
 - Inband trucking: AFM DR copies the data from the primary to the secondary using the DR transport protocol (NFSv3 or NSD). After the copy is complete, the primary and secondary have the exact same data and PSNAP0 is created.
 - Outband trucking: It is the user's responsibility to copy all the data from the primary and secondary by using any mechanism that they choose. For example, rsync with **mtime** preserved. After all the data is copied, the user can use the AFM DR commands to establish the DR relationship that creates the PSNAP0.

Note: Both methods use the **--inband** option of the **mmafmctl** command while setting up the relationship between the primary and secondary sites.

For more information about the two trucking methods, see *Converting GPFS fileset to AFM DR* in *IBM Spectrum Scale: Administration Guide*. See the following use cases for recommendation on when to use inband versus outband trucking.

AFM DR trucking use cases

Create the primary and secondary filesets from scratch

The primary and secondary clusters are empty and AFM DR establishes the relationship by creating PSNAP0 that is empty.

Trucking method recommendation: Use the inband trucking method for this scenario because the clusters have no data to copy and create PSNAP0.

Create a primary and secondary fileset from scratch, where primary has data and secondary does not have data

In this case, either the inband or the outband mechanism can be used to copy the data to the secondary cluster.

- Inband:

By using the inband method, the existing data in the primary is copied by AFM DR to the secondary. To copy this data a work queue is created on the gateway that contains all of the files that need to be copied and file system operations that need to be performed on the secondary. Then, the gateway processes this queue as quickly as possible by copying data to the secondary. When all of the work, which was outstanding when the command was run, is flushed to the secondary, a PSNAP0 is created. The trucking process ends when a snapshot is created.

It is recommended to start applications on the primary after the trucking process is complete. However, if applications are started while the trucking process is in progress, the gateway node queues the data generated by the application and the requests from the trucking process at the same time. The gateway node queue buffer is filled with many requests, which causes the queue memory overflow. In such a case, the queue is dropped and the gateway node goes into the recovery mode. This can result in a cycle where the primary cluster always lags behind the secondary cluster. The potential solutions are:

- Add bandwidth to handle the initial synchronization workload. This bandwidth can involve tuning – for example, increasing the number of flush threads to increase the gateway throughput, additional hardware or network bandwidth.
- Keep the application from writing to the primary until the trucking process is complete and PSNAP0 is created.
- Increase the gateway node memory allocation to prevent memory overflows.

- Outband:

In this case, a complete copy of the data exists before the primary-secondary relationship is created. The data can be copied to the secondary by using any method that includes a restore from a backup or a rsync process. When an identical copy of the data exists at the secondary, AFM DR can establish the DR relationship. The primary and secondary need to be identical when the relationship is established, so during this time the data must not change in the primary. After the PSNAP0 is created, applications can start by using the AFM DR filesets.

Note: Both methods use the **--inband** option of **mmafmctl** command while setting up the relationship between the primary and secondary sites. For data that is already copied to the secondary site, **mtime** of files at the primary and secondary site is checked. If **mtime** values of both files match, data is not copied again and existing data on the secondary site is used. If **mtime** values of both files do not match, existing data on the secondary site is discarded and data from the primary site is written to the secondary site.

The primary fileset is empty and the secondary fileset has data

Trucking method recommendation: In such a situation, the recommended mechanism is outband trucking. The administrator copies data from the secondary to the primary and then uses the AFM DR conversion mechanism to establish the DR relationship.

Data exists on the 'to-be' primary and the secondary filesets

Ensure that the data is identical on the primary and the secondary before the AFM DR conversion is performed.

Trucking method recommendation: This can be done by using only the outband trucking. AFM DR checks whether the data is the same when the relationship is established.

Failover and failback

Failover

After a failure of the primary, applications can fail over to the secondary fileset. Performing a failover operation on the secondary fileset converts it to the primary and makes it writable. AFM DR failover has two modes: “As is” and “Revert to the latest RPO snapshot on the secondary”:

- As is:

The “as is” data corresponds to the last byte of data that was copied to the secondary cluster before the primary cluster failed. This is always the latest data available on the secondary cluster. It is the recommended option and also the default behavior.

- Revert to the latest RPO snapshot on the secondary:

Restoring data from the last RPO snapshot results in data being restored from the past. As a result, this process is not recommended in AFM DR as it might lead to data loss. For example, although the RPO interval was set to 24 hours, the failover occurred at the 23rd hour, then restoring from the last snapshot (taken 24 hours ago) can result all the data that is created in the last 23 hours being lost. This failover option can take extra time as the data from the snapshot must be restored completely before the application can start by using the fileset.

Failback

There are two options in AFM DR to failback to the original primary:

- Failback Option 1: New primary

Create the primary from scratch because all the data in the primary was lost. To create a primary from scratch, all the data needs to be copied to the original primary from the active primary by using whichever mechanism the administrator chooses.

- Fail back Option 2: Temporary loss of primary

A temporary failure of the primary that results in a failover of the application to the secondary. To reestablish the original primary:

AFM DR uses the last RPO snapshot to calculate exactly what data changed on the secondary since the failover occurred. So for this to work there needs to be a common snapshot between the primary and secondary. The following steps give a simplified illustration of such a scenario:

1. A PSNAP was created between the primary and secondary at 8 AM. At 8 AM, the snapshot in the primary and secondary have the exact same data.
2. At 9 AM, the primary fails.
3. At 9 AM, the applications failover to secondary.
4. Primary is back up at 10 AM that is 2 hours later.
5. Restore the T0 snapshot on the primary fileset.
6. Restore the 8 AM snapshot on the primary fileset.
7. Copy the changed data that is calculated in step 6 from the secondary fileset to the primary fileset. Now, the primary is in sync with the secondary fileset.

Note: For more information about the failback, see *Failback procedures* in the *IBM Spectrum Scale Administration Guide*.

For a temporary primary failure, the PSNAP (peer-to-peer snapshot) mechanism can be used to periodically take synchronized snapshot between the primary and the secondary. The snapshots can be scheduled at fairly large intervals, for example, one or two times a day. Only the data created in the secondary since the last PSNAP needs to be copied back to the primary during a failback.

Note:

The amount of data copied back to the primary depends on the following:

- How long the primary is unavailable.
- The amount of data that is created since the primary failure.
- How long the failure was from the last snapshot. For example, if the RPO snapshot is captured every 24 hours and the primary fails at the 23rd hour, then changes for the last 23 hours need to be recovered.

Deployment considerations for AFM DR

See the following considerations to understand if AFM DR fulfills your production requirements for disaster recovery:

Amount of data created per hour or per day

With data replication the amount of data created in a day determines the network bandwidth and the gateway node design, and dictates whether the replication rates can be supported by AFM DR. You can determine the requirements by looking at:

- Network bandwidth:

The network must have the bandwidth to accommodate a transfer rate equivalent to the rate data is generated. You should consider how you want to handle high traffic events like fail-back where I/O requirements can be greatly increased for a short period of time.

- Gateway nodes(s)

A gateway tracks all the changes and replicated the data from the primary to the secondary. As the number of filesets increase you can increase the number of gateway nodes in the primary cluster to increase throughput. The number of gateway nodes needed depends on:

- The number of primary filesets.
- The rate of data changes generated by each fileset.
- The bandwidth of the gateway hardware and the network connections between clusters.
- The method in which the filesets are distributed across gateway nodes. For example, it is possible that all the heavily loaded filesets are allocated to the same gateway node.

Note: There is no manual way of controlling the fileset allocation to a gateway node. Currently, the method in which AFM DR allocates fileset to the gateway node is ad hoc and creates another challenge for the user.

The number of filesets on the primary cluster

The number of filesets impact the following factors:

- You might need to increase the number of gateway nodes for even distribution of the workload.
- The AFM DR RPO mechanism creates and deletes a snapshot after every RPO interval. As the number of filesets increase, creating the RPO snapshots simultaneously for all the filesets might cause a significant load on the system. The RPO can be disabled if there is no need to failback after a temporary primary failure. Alternatively, setting a large RPO interval can relieve some of this pressure however, once the RPO is enabled then all of the above points need to be considered carefully.

Note: A gateway node failure and recovery causes redistribution of the fileset workload, which in turn might cause changes in the performance of the replication characteristics.

Best Practices for AFM DR

Minimizing data loss during primary failure

AFM DR is asynchronous, so there is always a chance data can be lost in the event of a failure. The amount of data lost is affected by:

- The network bandwidth between the primary and secondary.

- The performance of the gateway which depends on IBM Spectrum Scale tuning, the amount of memory and CPU available to gateway node and the number of filesets allocated to each gateway node. If the gateway node is overloaded, it can result in replication rate reduction and reduced network bandwidth utilization.
- The ability of the primary and secondary to read and write the data to disk.

The AFM DR replication should be tuned to minimize the data loss in case of primary failure by keeping-up with the data creation rate. Some workloads occur in bursts, in this case you can design for average data transfer rates though keep in mind the time to sync is much greater right after a burst of changes.

Note: The AFM DR replication rate is independent of the RPO interval. The RPO interval does not affect the data loss sustained during primary failure.

Generating notification for failed replication

There is no automated notification mechanism in AFM DR to monitor the replication rates that are falling behind as long as they are within the RPO. However, a script can be written to periodically test the gateway node to see how fast the message queue is being processed. This provides an estimate of the replication rate sustained by the gateway node.

To monitor the RPO you can use the AFMRPOMISS callback event. This event is triggered if the RPO snapshot is not taken at the set interval. The event indicates something is wrong within the system, and can be used as a trigger to start an analysis of what needs to be rectified within the system to bring it back to optimal performance.

Using tuning parameters to improve performance

There are several AFM DR tuning parameters that can be used to tune performance. For more information on tuning parameters, see *Configuration parameters for AFM And AFM-DR* in the *IBM Spectrum Scale: Administration Guide*.

AFM DR use case

This use case describes the AFM-based Asynchronous Disaster Recovery (AFM DR) process when NFS and SMB clients are mounted from the primary IBM Spectrum Scale cluster.

Overview of AFM DR when NFS and SMB clients are mounted from the primary cluster

During scheduled maintenance on the primary cluster or a failure at the primary site cluster, those NFS, and SMB clients that mount exports from the primary cluster experience a brief and temporary outage, when the **mmafmctl failover** commands are run. After the **mmafmctl failover** commands complete, the NFS and SMB clients can mount from the new acting primary cluster (formerly the secondary cluster).

Note: To ensure a successful failover, system administrators at each site need to create identical export definitions on the secondary cluster. It is important to keep the definitions up to date when changes occur on the primary cluster.

After you create initial AFM DR relationships, you can create NFS and SMB exports so that applications can access the IBM Spectrum Scale data via protocols.

Use case 1: The path to the data is identical on both the primary and secondary clusters

1. On both the primary and the secondary clusters, enter the same path and client (-c) definitions as shown in the following example:

```
mmnfs export add /ibm/gpfs0/export1 -c
9.11.102.0/24(Access_Type=R0,Squash=root_squash;
9.11.136.0/23(Access_Type=R0,Squash=root_squash;
```

```
10.18.40.0/22(Access_Type=RW,Squash=root_squash;  
10.18 48.0/22(Access_Type=RW,Squash=root_squash;
```

2. On the NFS clients,

- Issue the **showmount -e** command and the IP address of the primary cluster as shown in the following example:

```
showmount -e 10.18.96.100  
Export list for 10.18.96.100:  
/ibm/gpfs0/export1  
9.11.102.0,911.136.0,10.1840.0,10.18.48.0
```

- Issue the **showmount -e** command and the IP address of the secondary cluster as shown in the following example:

```
showmount -e 10.18.50.30  
Export list for 10.18.50.30:  
/ibm/gpfs0/export1  
9.11.102.0,911.136.0,10.18.40.0,10.18.48.0
```

3. On the NFS clients, issue the following command to mount from the primary CES IP addresses as shown in the following example:

```
mount -o hard, sync  
10/18.96.100:/ibm/gpfs0/export0 /mnt_pnt
```

4. If a failure occurs on the primary cluster, initiate failover.

5. On the NFS clients, issue the following commands to unmount from the primary CES IP addresses and then mount from the secondary CES IP addresses as shown in the following example:

```
umount -f /mnt_pnt  
mount -o hard, sync  
10.18.50.30:/ibm/gpfs0/export0 /mnt_pnt
```

Use case 2: The path to the data is different on the primary and secondary cluster

1. Create the export with the same client (-c) definitions.

- Issue the following command and the IP address of the primary cluster:

```
mmnfs export add /ibm/gpfs0/export1 -c  
9.11.102.0/24(Access_Type=R0,Squash=root_squash;  
9.11.136.0/23(Access_Type=R0,Squash=root_squash;  
10.18.40.0/22(Access_Type=RW,Squash=root_squash;  
10.18 48.0/22(Access_Type=RW,Squash=root_squash;
```

- Issue the following command and the IP address of the secondary cluster:

```
mmnfs export add /ibm/gpfs3/export9 -c  
9.11.102.0/24(Access_Type=R0,Squash=root_squash;  
9.11.136.0/23(Access_Type=R0,Squash=root_squash;  
10.18.40.0/22(Access_Type=RW,Squash=root_squash;  
10.18.48.0/22(Access_Type=RW,Squash=root_squash;
```

2. On the NFS clients,

- Issue the **showmount -e** command and the IP address of the primary cluster:

```
showmount -e 10.18.96.100  
Export list for 10.18.96.100:  
/ibm/gpfs0/export1  
9.11.102.0,911.136.0,10.18.40.0,10.18.48.0
```

- Issue the **showmount -e** command and the IP address of the secondary cluster:

```
showmount -e 10.18.50.30  
Export list for 10.18.50.30:
```

```
/ibm/gpfs3/export9  
9.11.102.0,911.136.0,10.18.40.0,10.18.48.0
```

3. On the NFS clients, issue the following command to mount from the primary CES IP addresses:

```
mount -o hard, sync  
10.18.96.100:/ibm/gpfs0/export0 /mnt_pnt
```

4. On the NFS clients, issue the following commands to unmount from the primary CES IP addresses and then mount from the secondary CES IP addresses:

```
umount -f /mnt_pnt  
mount -o hard, sync  
10.18.50.30:/ibm/gpfs3/export9 /mnt_pnt
```

AFM to cloud object storage

The AFM to cloud object storage is an IBM Spectrum Scale feature that enables placement of files or objects in an IBM Spectrum Scale cluster to a cloud object storage.

Cloud object services such as Amazon S3, IBM Cloud® Object Storage, Seagate Lyve Cloud, and Microsoft Azure Blob storage offer industry-leading scalability, data availability, security, and performance. AFM to cloud object storage supports Amazon S3, IBM Cloud Object Storage, Seagate Lyve Cloud, Microsoft Azure Blob storage, and Google Cloud Platform. Azure Blob storage is supported by deploying MinIO as a S3 gateway between AFM to cloud object storage and Azure Blob storage. The AFM to cloud object storage allows associating an IBM Spectrum Scale fileset with a cloud object storage. Customers use a cloud object storage to run workloads such as mobile applications, backup and restore, enterprise applications, and big data analytics, file server. These workloads can be cached on AFM to cloud object storage filesets for faster computation and synchronize back to the cloud object storage server.

Microsoft Azure Blob Storage doesn't have native support for the S3 API. MinIO can be used to convert S3 APIs to Azure Blob native. AFM to cloud object storage communicate to MinIO Gateway which in-turn talk to Azure Blob backend storage.

MinIO Blob Storage Gateway (S3 API) is also available as a 'Fully-Managed' application on Azure marketplace.

The front-end for object applications is an AFM to cloud object storage fileset with the data exchange between the fileset and cloud object storage buckets through the AFM to cloud object storage in the background by providing high performance for the object applications. Object applications can also span across AFM to cloud object storage filesets and on a cloud object storage. Both the fileset and the cloud object storage can be used as a backup of important data.

The following figure illustrates the AFM to cloud object storage:

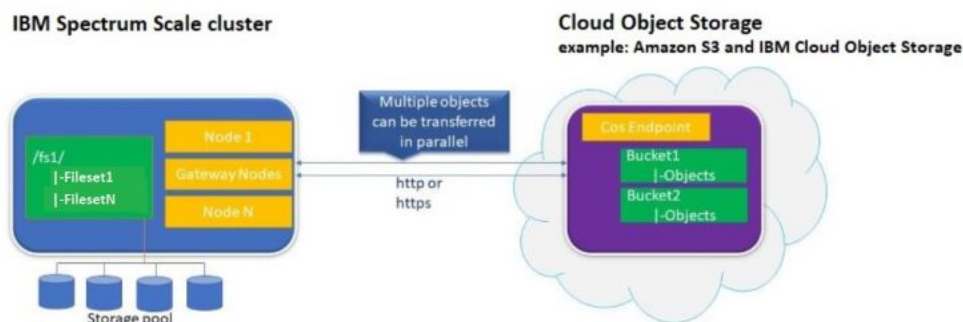


Figure 16. AFM to cloud object storage

The AFM to cloud object storage on an IBM Spectrum Scale fileset becomes an extension of cloud object storage buckets for high-performance or used objects. Depending upon the modes of AFM to cloud object storage fileset configurations, objects required for applications such as AI and big data analytics can be downloaded, worked upon, and can be uploaded to a cloud object storage. The objects that are created

by applications can be synchronized to the objects on a cloud object storage asynchronously. An AFM to cloud object storage fileset can cache only metadata or both metadata and data.

The AFM to cloud object storage also allows data center administrators to free the IBM Spectrum Scale storage capacity by moving less useful data to the cloud storage. This feature reduces capital and operational expenditures. The AFM-based cache eviction feature can be used to improve the storage capacity manually and by using policies. For more information about the AFM cache eviction, see [“Cache eviction”](#) on page 73.

The AFM to cloud object storage uses the same underlying infrastructure as AFM. For more information, see [“Active File Management”](#) on page 38.

The AFM to cloud object storage is available on all IBM Spectrum Scale editions.

AFM to cloud object storage by using Azure Blob storage

AFM to cloud object storage fileset can be configured to use Microsoft Azure Blob storage backend by deploying MinIO in between. MinIO runs as an S3 Gateway, and it converts S3 API to/from Azure Blob native. All communication between AFM to cloud object storage fileset and the bucket at the Azure Blob Storage goes through S3 Gateway. All the operations, modes and functions are seamlessly supported with S3 Gateway that communicates to Azure Blob.

The following figure illustrates the AFM to cloud object storage by using Azure Blob as backend through MinIO S3 gateway:

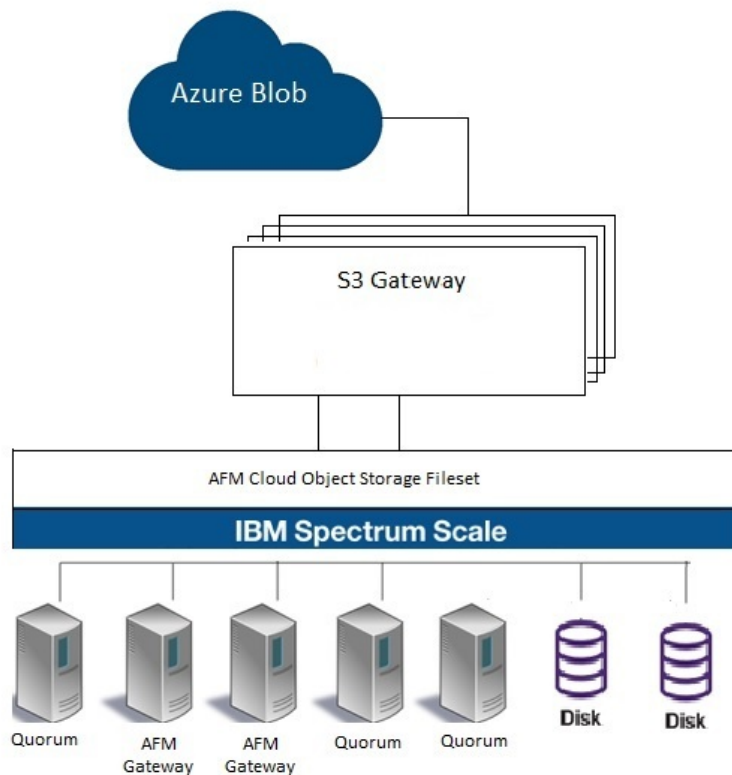


Figure 17. AFM to cloud object storage by using Azure Blob

AFM to cloud object storage operation modes

AFM to cloud object storage filesets and objects have different operation modes.

AFM to cloud object storage operations are serviced either synchronously or asynchronously. Reads and revalidations are synchronous operations, and update operations from an AFM to cloud object storage fileset are asynchronous to a cloud object storage depending upon the **afmAsyncDelay** interval. All update operations from the writable AFM to cloud object storage filesets such as IW or SW mode are on the primary gateway. Queues on the primary gateway are pushed to the cloud object storage asynchronously based on the **afmAsyncDelay** interval.

Fileset operation modes

An AFM to cloud object storage fileset is supported on all existing AFM fileset modes. All existing AFM modes behave in a similar way with AFM to cloud object storage except the local-update (LU) mode.

Unlike AFM LU-mode, where data cannot be pushed to the home, AFM to cloud object storage LU-mode fileset can upload data to the target bucket of the cloud object storage server by using the **mmafmcscctl upload** command. The upload operation is a manual operation. You can use this operation to run applications in the LU-mode cache and synchronize the data to the bucket.

Single writer (SW)

In this mode, only an AFM to cloud object storage fileset does all the writing and this fileset does not check a cloud object storage for object updates. The AFM to cloud object storage fileset does not enforce writing or modifying an object on a cloud object storage. After an SW-mode AFM to cloud object storage fileset is created, if a cloud object storage has pre-existing data, this data can be prefetched by using the `download` option from the **mmafmcscctl** command.

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 Spectrum Scale cluster or on a different cluster and points to a cloud object storage. There is no synchronous locking between clusters while updating objects on a cloud object storage. Each AFM to cloud object storage fileset reads from a cloud object storage and makes updates to the cloud object storage independently based on the revalidation intervals and asynchronous delay.

This mode is used to access different objects from each IW AFM to cloud object storage site, for example, unique users at each site that update objects in their cloud object storage bucket. This mode allows multiple AFM to cloud object storage clusters to modify the same set of objects. Only advanced users must do this modification because there is no locking or ordering between updates. Updates are propagated to a cloud object storage 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 site to be undetermined.

Read-only (RO)

In this mode, data in the AFM to cloud object storage fileset is read-only. You cannot create or modify files in an RO-mode AFM to cloud object storage fileset. If an application is using an object while it is being re-created, that is, deleted and re-created with the same name on a cloud object storage, the object gets re-created in AFM to cloud object storage fileset. The **mmafmcscctl download** command can be used to download the objects on priority or preference.

Local updates (LU)

This mode is similar to the RO mode, although you can create and modify objects in an AFM to cloud object storage fileset. Updates in the AFM to cloud object storage fileset are considered local to the AFM to cloud object storage fileset and get decoupled from the corresponding object on a cloud object storage. Local updates are never pushed back to a cloud object storage.

When an object is downloaded and modified in an AFM to cloud object storage LU-mode fileset, the modified data does not synchronize back with the bucket on the cloud object storage server even if the object on the bucket is modified. This behavior is similar to the AFM LU-mode fileset. For more information about AFM modes, see [“Operations with AFM modes” on page 48](#).

Behaviors with local objects

In AFM, LU-mode objects have one of the following states:

Uncached

Objects on a cloud object storage are shown as uncached. For these objects, only metadata is copied into an AFM to cloud object storage fileset. The object does not reside on the AFM to cloud object storage fileset, but only on the cloud object storage. Changes on the cloud object storage are reflected in the AFM to cloud object storage fileset.

Cached

If an uncached object is read in an AFM to cloud object storage fileset or pre-fetched, the state of the object changes to cached. In the cached state, all changes to the object on a cloud object storage are reflected in the AFM to cloud object storage fileset. The object resides on the AFM to cloud object storage fileset.

When symlinks are created on an AFM to cloud object storage cache fileset, the relationship between the symlink file and the target file is maintained and the files are uploaded to the bucket.

The symlinks relationship is maintained only if they are created in an AFM to cloud object storage fileset. If the symlinks relationship is not created in the AFM to cloud object fileset, it is not maintained.

Local

Object data or metadata that is modified on an AFM to cloud object storage fileset becomes local to the AFM to cloud object storage fileset. The cached objects relationship to the object in a cloud object storage is broken. Changes on the cloud object storage are not reflected in the AFM to cloud object storage fileset anymore and object changes are not copied to the cloud object storage.

Note: Objects can be downloaded from a cloud object storage and uploaded back without affecting the LU-mode. When the object is downloaded, it is synced-in to the object on the cloud object storage.

Manual updates (MU)

The manual update (MU) mode supports manual replication of the files or objects by using ILM policies or user provided object list. MU mode is supported on AFM to cloud object storage backend. The IBM Spectrum Scale independent fileset can be converted to MU mode fileset or MU mode fileset can be created as a new relationship to cloud object storage.

MU mode fileset provides the flexibility to upload and download files or objects to and from cloud object storage after you finalize the set of objects to upload or download. Unlike other AFM to cloud object storage object fileset modes, MU mode depends on manual intervention from administrators to upload and download the data to be in sync. As administrators you can also automate upload and download by using ILM policies to search specific files or objects to upload or download.

MU mode semantics:

- Applications or users create data in the MU mode fileset. The data is not transferred to cloud object storage backend automatically. The data can be dirty or hot data.
- Either all the files or specific files can be determined and uploaded by using the **mmafmcscctl upload** command. Or, policies can help determine these specific files.
- When the files are added to cloud object storage or the data is changed on cloud objects storage, these files can be downloaded using **mmafmcscctl download** command.

Important: MU mode does not recognize the files added to cloud object storage or modified data automatically, to download these files or data administrators can create object list that contains the names of the files from cloud object storage.

- Metadata is refreshed only once. This is applicable only if the MU fileset is created pointing to non-empty bucket. Each file is refreshed before allowing the update if **readdir** was not performed.
- All the changes to files are local to the fileset except remove or rename operations which can be made automatically replicated by using the **config** option.

Deleting files from MU mode fileset

Files on MU mode fileset take up inode and space from fileset. Files can be deleted from MU mode fileset but the file inodes are reclaimed from the fileset inode space only when **mmafmcscctl delete** commands are issued. MU mode has two delete options backed by **mmafmcscctl delete** command. The files can be deleted locally by using **posix** commands. For example, **rm**. MU mode does not reclaim inodes immediately after the deletion until the delete operation is executed to the COS target.

The **mmafmcscctl delete** commands – from cache option looks for deleted files in the MU fileset and reclaim the file inodes, whereas –from-target option looks for the deleted files in MU fileset, reclaims the inodes and queue the delete operation to COS so that files deleted from MU fileset gets deleted on COS as well.

Note: After the delete --from-cache is performed, the inodes are reclaimed from MU mode fileset. After execution of this command if the delete --from-target is executed, the deletes are not queued on the cloud object storage.

Example : Deleting and reclaiming inodes from MU mode fileset

1. Create MU mode fileset.

```
node1 1] mmafmcosconfig fs1 mufilesetdemo --endpoint http://s3.us-east.cloud-object-
storage.appdomain.cloud --bucket demobucketx --object-fs --mode mu
node1 1]
node1 1]
node1 1] mmafmctl fs1 getstate
Fileset Name      Fileset Target      Cache State
Gateway Node      Queue Length      Queue numExec
-----
mufilesetdemo     http://s3.us-east.cloud-object-storage.appdomain.cloud:80/demobucketx
Inactive
node1 1]
```

2. Check the used inodes.

```
node1 24:16 1] mmlsfileset fs1 mufilesetdemo -i
Collecting fileset usage information ...
Filesets in file system 'fs1':
Name      Status      Path
InodeSpace MaxInodes AllocInodes UsedInodes
mufilesetdemo Linked      /gpfs/fs1/mufilesetdemo
3          100352     100352     3
node1 1]
```

3. Create five files.

```
node1 1] echo "filedata" > /gpfs/fs1/mufilesetdemo/file1
node1 1] echo "filedata" > /gpfs/fs1/mufilesetdemo/file2
node1 1] echo "filedata" > /gpfs/fs1/mufilesetdemo/file3
node1 1] echo "filedata" > /gpfs/fs1/mufilesetdemo/file4
node1 1] echo "filedata" > /gpfs/fs1/mufilesetdemo/file5
```

4. Check inode stats : Here inode numbers are increased by five.

```
node1 1] mmlsfileset fs1 mufilesetdemo -i
Collecting fileset usage information ...
Filesets in file system 'fs1':
Name      Status      Path
InodeSpace MaxInodes AllocInodes UsedInodes
mufilesetdemo Linked      /gpfs/fs1/mufilesetdemo
3          100352     100352     8
node1 1]
```

5. Create a object list with all files and upload.

```
node1 1] cat /root/objectlist
/gpfs/fs1/mufilesetdemo/file1
/gpfs/fs1/mufilesetdemo/file2
/gpfs/fs1/mufilesetdemo/file3
/gpfs/fs1/mufilesetdemo/file4
/gpfs/fs1/mufilesetdemo/file5
node1 1]
node1 1]
node1 1] mmafmcosctl fs1 mufilesetdemo /gpfs/fs1/mufilesetdemo upload --object-list /
root/objectlist
Queued      (Total)      Failed      TotalData
5           (5)           0           (approx in Bytes)
45
Object Upload successfully queued at the gateway.
node1 1]
```

6. Remove all the files from the MU fileset.

```
node1 1] rm -rf /gpfs/fs1/mufilesetdemo/file*
node1 1]
```

7. Check the inodes stats : inodes are not yet reclaimed from MU fileset.

```
node1 1] mmlsfileset fs1 mufilesetdemo -i
Collecting fileset usage information ...
Filesets in file system 'fs1':
Name          Status    Path
InodeSpace    MaxInodes  AllocInodes  UsedInodes
mufilesetdemo Linked      /gpfs/fs1/mufilesetdemo
3             100352     100352       8
node1 1]
```

8. Issue the following command to reclaim the inodes from MU mode fileset:

```
node1 1] mmafmcosctl fs1 mufilesetdemo /gpfs/fs1/mufilesetdemo/ delete --from-cache
```

9. Check the inode stats : Here the used inodes are reclaimed.

```
node1 1] mmlsfileset fs1 mufilesetdemo -i
Collecting fileset usage information ...
Filesets in file system 'fs1':
Name          Status    Path
InodeSpace    MaxInodes  AllocInodes  UsedInodes
mufilesetdemo Linked      /gpfs/fs1/mufilesetdemo
3             100352     100352       3
node1 1]
```

Note: The **mmafmcscctl** command with option **delete --from-cache**, does not queue deletes to COS.

Example : Deleting files, reclaiming inode, and queueing deletes to COS

1. Check MU mode fileset inode stats.

```
Node1 1] mmlsfileset fs1 mufilesetdemo -i
Collecting fileset usage information ...
Filesets in file system 'fs1':
Name          Status    Path
InodeSpace    MaxInodes  AllocInodes  UsedInodes
mufilesetdemo Linked      /gpfs/fs1/mufilesetdemo
3             100352     100352       3
Node1 1]
```

2. Create five files in fileset1.

```
Node1 1] echo "filedata" > /gpfs/fs1/mufilesetdemo/file1
Node1 1] echo "filedata" > /gpfs/fs1/mufilesetdemo/file2
Node1 1] echo "filedata" > /gpfs/fs1/mufilesetdemo/file3
Node1 1] echo "filedata" > /gpfs/fs1/mufilesetdemo/file4
Node1 1] echo "filedata" > /gpfs/fs1/mufilesetdemo/file5
Node1 1]
```

3. Check the inodes stats:Used inodes increased by five.

```
Node1 1] mmlsfileset fs1 mufilesetdemo -i
Collecting fileset usage information ...
Filesets in file system 'fs1':
Name          Status    Path
InodeSpace    MaxInodes  AllocInodes  UsedInodes
mufilesetdemo Linked      /gpfs/fs1/mufilesetdemo
3             100352     100352       8
Node1 1]
```

4. Create a object list and upload all the files to COS.

```
Node1 1]
Node1 1] mmafmcosctl fs1 mufilesetdemo /gpfs/fs1/mufilesetdemo upload --object-
list /root/objectlist
Queued      (Total)      Failed      TotalData
              (5)              0              (approx in Bytes)
              5              45
Object Upload successfully queued at the gateway.
Node1 1]
```

5. Remove the files from MU mode fileset.

```
Node1 1] rm -rf /gpfs/fs1/mufilesetdemo/file*
Node1 1]
```

6. Check the inode usage : Inodes are not reclaimed after delete.

```
Node1 1] mmlsfilesset fs1 mufilesetdemo -i
Collecting filesset usage information ...
Filesets in file system 'fs1':
Name      Status      Path
InodeSpace MaxInodes AllocInodes UsedInodes
mufilesetdemo Linked /gpfs/fs1/mufilesetdemo
3          100352    100352      8
Node1 1]
```

7. Issue **mmafmcctl** delete command with **-from-target** option.

```
Node1 1] mmafmcctl fs1 mufilesetdemo /gpfs/fs1/mufilesetdemo/ delete --from-target
```

8. Check that the remove operations are queued and executed to cos.

```
Node1 1] mmafmctl fs1 getstate
Fileset Name      Fileset Target      Cache State
Gateway Node      Queue Length      Queue numExec
-----
mufilesetdemo     http://s3.us-east.cloud-object-storage.appdomain.cloud:80/demobucketx
Dirty c7f2n04 5    22
Node1 1] mmafmctl fs1 getstate
Fileset Name      Fileset Target      Cache State
Gateway Node      Queue Length      Queue numExec
-----
mufilesetdemo     http://s3.us-east.cloud-object-storage.appdomain.cloud:80/demobucketx
Active c7f2n04 0    27
Node1 1]
```

9. Check that the inodes in MU fileset are reclaimed.

```
Node1 1] mmlsfilesset fs1 mufilesetdemo -i
Collecting filesset usage information ...
Filesets in file system 'fs1':
Name      Status      Path
InodeSpace MaxInodes AllocInodes UsedInodes
mufilesetdemo Linked /gpfs/fs1/mufilesetdemo
3          100352    100352      3
Node1 1]
```

MU mode autoremove

To keep the deletes in sync with COS, the autoremove feature can be configured on the MU fileset by using **mmchfilesset** command. When the parameter **afmMUAutoRemove** is set to yes then removes are automatically queued to the cloud object storage.

For setting this **afmMUAutoRemove** parameter option first, do the following:

1. Stop the fileset by using **mmafmctl** stop option.
2. Set the parameter on the fileset by using **mmchfilesset <fs> <fileset> -p afmMuAutoRemove=yes**.
3. Again start the fileset by using **mmafmctl** start command.

Note: When the auto remove option is configured the inodes are reclaimed from MU mode fileset automatically.

Use case scenarios for MU mode:

1. The MU mode supports scenarios where the data changes at AFM to cloud object storage fileset is not needed to be in sync with the target cloud object storage. Applications can perform data modification at fileset level and whenever it is necessary to sync the data to the cloud object storage that can be synced.
2. Similarly, application running on cloud can modify the data on cloud and when it is required to be in sync with AFM to cloud object storage MU mode fileset, the data can be downloaded. This

helps in bandwidth utilization as data can be uploaded or downloaded at specific times. For example, application works on files in the MU fileset and uploaded to COS at later time in the day when the load on the network is less.

Object operation modes

AFM to cloud object storage supports **ObjectFS** and **ObjectOnly** behaviors.

These two behaviors are defined during the creation of a fileset by using the **mmafmcconfig --object-fs** command. If you specify the **--object-fs** option while you create an AFM to cloud object storage fileset, the **ObjectFS** mode behavior is enabled. If you skip the **--object-fs** option, an AFM to cloud object storage fileset is created with the **ObjectOnly** mode. The behavior is determined based on the workload requirement. When a mode is enabled, you cannot change it later.

objectFS mode

In the **objectFS** mode, an AFM to cloud object storage fileset is synchronized with a cloud object storage. AFM to cloud object storage RO, LU, and IW modes of filesets synchronize metadata to and from a cloud object storage. It includes operations such as **readdir** and **lookups** to synchronize the data. Objects are downloaded when they are read on demand or worked upon by an application that is running on an AFM to cloud object storage fileset. The **objectfs** enabled AFM to cloud object storage fileset behaves like an AFM fileset. For SW and IW modes-enabled AFM to cloud object storage fileset, the AFM to cloud object storage uploads files as objects to the object storage server. For an AFM RO, LU, and IW modes-enabled AFM to cloud object storage fileset, AFM automatically synchronizes objects from the cloud object storage to the AFM to cloud object storage fileset as files. Enable this parameter if the AFM to cloud object storage fileset behaves like an AFM modes fileset.

ObjectOnly mode

If the **object-fs** parameter is not defined, the **ObjectOnly** parameter will get enabled. This behavior is set on the AFM to cloud object storage filesets by default.

With the **ObjectOnly** mode, refresh of an AFM to cloud object storage fileset (AFM RO, LU, and IW mode fileset) with a cloud object storage will not be on-demand or frequent. You need to manually download data or metadata from the cloud object storage to the AFM to cloud object storage fileset, meanwhile data transfer from the AFM to cloud object storage fileset to the cloud object storage works automatically without manual intervention.

Enable this parameter on an AFM to cloud object storage fileset to avoid frequent trips and reduce the network contention by selectively performing the download or upload operations.

AFM to cloud object storage parallel read data transfer

Parallel read data transfer improves the overall data read transfer performance of an AFM to cloud object storage fileset by using multiple gateway nodes. When this feature is enabled, objects that are stored on a cloud object storage can be fetched in an IBM Spectrum Scale cluster fileset by using multiple gateway nodes.

To help the primary gateway read large files from the cloud object storage, an AFM cache cluster can be configured to use multiple gateways. On a cloud object storage, the same endpoints can point to a bucket or can set up in the distributed mode, that is, multiple servers by using same storage paths. Cloud object storage servers also provide multi-threaded multi-part download to achieve a maximum read performance.

In an AFM cache, gateway nodes can be mapped to endpoints of cloud object storage services. An export map replaces the **--endpoint** server name in the **mmafmcconfig** command to set up parallel reads. An export map can be changed without modifying the **--endpoint** option for a fileset. However, the fileset must be relinked or the file system must be remounted for the mapping to take effect. For more information about how to define, display, delete, and update mappings, see the *mmafmconfig* command in *IBM Spectrum Scale: Command and Programming Reference*.

To define and enable the parallel data read transfer for the AFM to cloud object storage, complete the following steps:

1. Define the mapping by using the **mmafmconfig** command.

A gateway can be defined manually by setting the **afmGateway** parameter. For more information, see *mmchfiles* command, in *IBM Spectrum Scale: Command and Programming Reference*.

2. Set up the keys for the target buckets by using the **mmafmcosskeys** command.
3. Use the mapping as an **--endpoint** when you set the AFM to cloud object storage relationship by using the **mmafmcossconfig** command.
4. Define the parallel read parameters. For more information about the parameters, see *Configuration parameters for AFM and AFM-DR* in *IBM Spectrum Scale: Administration Guide*.

All gateway nodes other than the primary gateway that is defined in a mapping are called participating or helper gateway nodes. The primary gateway of an AFM to cloud object storage fileset communicates with each of the participating gateway nodes, depending on their availability. When parallel data read transfer is configured, a single data transfer request is split into multiple chunks. Chunks are sent across to the participating gateway nodes in parallel for the transfer from a cloud object storage by using the respective mapping. The primary gateway processes the replies from all the participating gateway nodes, handles all data transfer failures, and coordinates activities until all data transfer is completed. If any participating gateway node fails, the primary gateway attempts to retry the failed task on the next available gateway and generates an error message in the *mmfs* log.

Note: For parallel data transfers, the gateway nodes in the mapping must have the same architecture and Linux operating system.

Parallel reads are effective on files or objects with sizes larger than files or objects that are specified by the parallel threshold. The threshold is defined by using the **afmParallelReadThreshold** parameter and is true for all types of objects and files.

Use the **afmParallelReadChunkSize** parameter to configure the size of each chunk. This parameter defines the minimum chunk size of the read that needs to be distributed among the gateway nodes during parallel reads. A zero value disables the parallel reads across multiple gateways.

Example

1. Create an export map by using multiple gateway nodes.

```
# mmafmconfig add map1_cos --export-map lb1.ait.cleversafelabs.com/  
c7f2n03,lb1.ait.cleversafelabs.com/c7f2n04  
--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.
```

Note: When the **--no-server-resolution** option is specified, AFM skips the DNS resolution of a specified hostname and creates the mapping. The specified hostname in the mapping must not be replaced with an IP address. Ensure that the specified hostname is resolvable when the mapping is in operation. This resolution is important if an endpoint resolves to multiple addresses and does not bind to a single IP address.

```
# mmafmconfig show map1_cos  
Map name: map1_cos  
Export server map: lb1.ait.cleversafelabs.com/c7f2n03,lb1.ait.cleversafelabs.com/c7f2n04
```

2. Set up an access key and a secret key for the bucket by using the export map.

```
# mmafmcoskeys cosbucket1:map1_cos set AccessKey SecretKey
```

3. Create an AFM to cloud object storage relation by using the **mmafmcossconfig** command.

```
# mmafmcossconfig fs1 cosbucket1 --endpoint http://map1_cos --object-fs --bucket  
cosbucket1 --debug --mode iw  
afmobjfs=fs1 fileset=cosbucket1  
bucket=cosbucket1 newbucket= objectfs=yes dir=  
policy= tmpdir= tmpfile= cleanup=yes mode=iw
```

```
xattr=no ssl=no autoRemove=no acls=no gcs=no vhb=
bucketName=cosbucket1 region= serverName=map1_cos cacheFsType=http
Linkpath=/gpfs/fs1/cosbucket1 target=http://map1_cos/cosbucket1
map=http://map1_cos cacheHost=map1_cos
endpoint=lb1.a1t.cleversafelabs.com cachePort=
endpoint=lb1.a1t.cleversafelabs.com ENDPOINT=--endpoint http://map1_cos
XOPT= -p afmParallelWriteChunkSize=0 -p afmParallelReadChunkSize=0
```

Here,

cosbucket1

An existing bucket with objects presents on the cloud object storage for reading.

--new-bucket

This option can be used to create a new bucket.

4. Tune the parallel transfer thresholds for parallel reads. The **afmParallelReadThreshold** parameter value is 1 GB and the **afmParallelReadChunkSize** parameter value is 512 MB.

```
# mmchfileset fs1 cosbucket1 -p afmParallelReadThreshold=1024
```

A sample output is as follows:

```
Fileset cosbucket1 changed.
```

```
# mmchfileset fs1 cosbucket1 -p afmParallelReadChunkSize=536870912
```

A sample output is as follows:

```
Fileset cosbucket1 changed.
```

Note: Set these parameters before you do any operation on the fileset to get these values in effect. If any operations are already done on the fileset, use the **mmafmctl stop** command, and then set these parameters and start the fileset.

5. Check that the values are set on the fileset.

```
# mmlsfileset fs1 cosbucket1 --afm -L
```

A sample output is as follows:

```
Filesets in file system 'fs1':

Attributes for fileset cosbucket1:
=====
Status                               Linked
Path                                /gpfs/fs1/cosbucket1
Id                                   5
Root inode                           1572867
Parent Id                             0
Created                               Thu Aug 19 15:24:39 2021
Comment
Inode space                           3
Maximum number of inodes              100352
Allocated inodes                      100352
Permission change flag                chmodAndSetacl
afm-associated                         Yes
Target                                http://map1_cos:80/cosbucket1
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              536870912
Parallel Read Threshold               1024
Number of Gateway Flush Threads       8
Prefetch Threshold                    0 (default)
Eviction Enabled                      yes (default)
Parallel Write Chunk Size             0
IO Flags                              0x0 (default)
```

6. List and read the object in the cache.

```
# ls -lash /gpfs/fs1/cosbucket1/object1
0 -rwxrwx--- 1 root root 2.0G Aug 19 2021 object1
```

7. Get all content of the object.

```
# cat /gpfs/fs1/cosbucket1/object1 > /dev/null
```

8. List and read again the object in the cache.

```
ls -lash /gpfs/fs1/cosbucket1/object1
2.0G -rwxrwx--- 1 root root 2.0G Aug 19 2021 object1
```

9. 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
cosbucket1	http://map1_cos:80/cosbucket1	Active	c7f2n03	0	10

10. To view the queue on the primary gateway and the helper gateway, use the following command:

```
# mmfsadm saferdump afm cosbucket1
```

A sample output is as follows:

```
Output snip -
Normal Queue: (listed by execution order) (state: Active)
0 Read [1572868.1572868] inflight (4194304 @ 12582912) tid 3454523
2 ReadSplit [1572868.1572868] inflight (486539264 @ 1610612736) tid 3401451
```

Here,

ReadSplit

Operations on gateways that are defined in the mapping can be seen.

For more information, see the *mmfsadm* command in *IBM Spectrum Scale: Problem Determination Guide*.

Connectivity to cloud object storage

Each AFM to cloud object storage file set in a cluster is served by one of the nodes that is designated as a gateway in the cluster.

The gateway node that is mapped to a fileset is called the primary gateway of the fileset. The primary gateway acts as the owner of the fileset and communicates with the cloud object storage by using the **mmafmtransfer** command. An IBM Spectrum Scale cluster that has many AFM to cloud object storage filesets can have multiple gateway nodes. These gateway nodes connect to the cloud object storage by using HTTP or HTTPS protocols.

Underlying protocol

Communication between IBM Spectrum Scale and a cloud object storage can be HTTP or HTTPS. A cloud object storage can be enabled with these two protocol choices.

The AFM to cloud object storage also enables sharing of data across AFM to cloud object storage clusters and a cloud object storage, even if the networks are unreliable or have high latency by using the built-in queuing optimization techniques of AFM. For more information, see [“Active File Management” on page 38](#).

Note:

Within a single IBM Spectrum Scale cluster with AFM to cloud object storage enabled, application nodes comply with POSIX semantics. The file or object locking across cache and a cloud object storage is not supported.

When you perform operations on AFM to cloud object storage filesets, ensure that the operations are supported on a cloud object storage over the chosen protocol. Because the operations that are performed from an AFM to cloud object storage fileset are replayed on the cloud object storage as normal object operations. Ensure that the bucket restrictions and limitations from a cloud object storage provider are followed.

The cache state of an AFM to cloud object storage is similar to an AFM or AFM-DR fileset. You can check the state by using the **mmafmctl getstate** command. This command is used to check the status of synchronization of the specified fileset. For example, if a bucket is not accessible on the cloud object storage server, the fileset is moved to the Unmounted state and the following log error is displayed:

```
2020-08-27_10:32:01.571-0400: [E] AFM: COS access of 192.168.1.1://bucket1 failed with error 13
for file system fs1 fileset ID 9. Caching will be disabled and the access will be tried again
after 300 seconds, on the next request to the gateway.
```

When the bucket is accessible, AFM to cloud object storage tries the synchronization again. Similarly, other states become valid with the AFM to cloud object storage fileset.

For more information about AFM fileset states, see *Monitoring AFM and AFM DR* in the *IBM Spectrum Scale: Problem Determination Guide*.

Eviction in AFM to cloud object storage

AFM to cloud object storage filesets use a storage space management process that is called eviction. There are two types of eviction methods, data eviction and metadata eviction.

Eviction is used to make space for new objects and files on an AFM to cloud object storage fileset. Eviction releases object or file data blocks (space) or its metadata (inode) in the fileset, if the fileset usage exceeds the fileset quota. This feature is useful when the storage that is provided by IBM Spectrum Scale is less than the cloud object storage. The quotas can be defined when you set the AFM to cloud object storage relationship by using the **mmafmcconfig** command with **--quota-files NumberOfFiles** and **--quota-blocks NumberOfBlocks** parameters.

The process of releasing blocks or inodes is called eviction. However, file data or metadata is not evicted if the file data is dirty. A file whose outstanding changes are not flushed to the home is called a dirty file. When files or objects are evicted from the cache, all data blocks of those files are cleared. The files or objects are uncached. When a read operation is performed on the files or objects, the files or objects fetch data from a cloud object storage. Similarly, any metadata operation such as lookup or ls populate the inodes that were evicted from the AFM to cloud object storage fileset.

While provisioning to control the number of inodes that are used by a cache or home fileset, you can specify a limit on the number of inodes for that fileset during the fileset creation or change the limits by using the **mmchfileset** command with the **--inode-limit** option. For more information, see *mmchfileset command* in the *IBM Spectrum Scale: Command and Programming Reference*.

For more information about the data and metadata eviction, see *Evicting files or objects data and Evicting files or objects metadata* in the *IBM Spectrum Scale: Administration Guide*.

AFM to cloud object storage limitations

The AFM to cloud object storage limitations are as follows:

- Hard links are supported only in the Local-Update (LU) mode. The creation of hard links fails with permission denied or E_PERM error on other modes.
- The primary resources of object servers are buckets and objects.
- Directory names are prefix to the objects. Empty directories are not replicated to the target object server.

- The ChangeTimes operation is supported on the cache filesets. However, this operation does not replicate to the target cloud object server.
- To replicate the chmod and chown metadata operations to a cloud object storage, you need to use the **mmafmcconfig** command with the **--xattr** option.
- The rename operation is not supported on a non-empty directory. When a non-empty directory is renamed, the rename operation fails with the E_NOTEMPTY error. However, empty directories and local directories can be renamed.
- Parallel data transfer for write operations is not supported on an AFM to cloud object storage enabled fileset. Objects are not synchronized by splitting chunks across gateway nodes but objects are queued on different gateway nodes based on the mapping.
- Deletion of objects on the cloud object storage synchronizes the deletion of files on the cache. But empty directories might remain on the cache.
- Support of file and directory naming convention is in accordance with the cloud object storage server supported guidelines. Some cloud object storage servers do not support special characters. The file or directory name must not contain special characters.
- File paths that have more characters than maximum characters limitation are not supported.
- You can run AFM to cloud object storage commands such as **mmafmckeys**, **mmafmcconfig**, **mmafmcctl**, and **mmafmcaccess** only from a Linux node.
- **Immutability** and **appendOnly** features are not supported on AFM to cloud object storage filesets.
- The **--iam-mode** option is not supported on AFM to cloud object storage filesets.
- If a file is evicted from AFM to cloud object storage fileset, the snapshot file of the evicted file contains zeros.
- AFM to cloud object storage supports Amazon S3 and IBM Cloud Object Storage.
- AFM to cloud object storage fileset supports setting an access control list (ACL) on a file of up to 2 KB size. When an ACL is assigned on a file of more than 2 KB size, the file is discarded and only file data is synchronized with the bucket.
- The symlinks that are created on an AFM to cloud object storage local mode (LU) fileset cannot be uploaded manually to the bucket.
- A dependent fileset linking is not supported in an AFM to cloud object storage fileset.
- In some cases, an AFM cache has pending changes such as delete and rename to replicate, and the fileset recovery is triggered. In this case, the cloud object storage might have old data and renamed files or directories are replicated. Because of the files or directories replication, some extra files might exist on the cloud object storage. The AFM cache and the cloud object storage have the latest copy of the files.
- AFM to cloud object storage is supported with Amazon S3, IBM Cloud Object Storage, and Microsoft Azure Blob storage. Azure Blob storage is supported by deploying a S3 gateway, like placing MinIO® between AFM to cloud object storage and Azure Blob storage.
- AFM to cloud object storage does not maintain sparseness of files when uploading the files to cloud object storage.
- If a file at AFM cache is already in sync with home or cloud object storage target and now renamed at cache, the next upload can leave duplicate objects at COS (both old and renamed version). You need to manually delete old objects from the COS.
- Creation of new bucket by using **mmafmcconfig --new-bucket** is not supported for Google Cloud Services. You must create bucket manually before you create AFM to cloud object storage fileset for Google GCP backend storage.
- Setting more than 2K metadata on AFM to cloud object storage filesets root is not supported.
- Based on the user defined policy, **reconcile** command will not upload empty directory or directory which has objects not meeting the policy criteria to cloud objects storage.
- The directory object feature is not supported when MinIO is used as a cloud object storage server.

ObjectOnly mode AFM to cloud object storage fileset support:

Upload or write and download or read operations are replicated to a target object server. The operations such as delete and rename are not synchronized between a cache and an object server.

- If data on a cache is deleted or renamed, the data is not synchronized to an object server.
- If an object on an object storage server is deleted or renamed, and the object is already replicated to a cache, the object is not synchronized to the cache.

Audit messages support for the AFM to cloud object storage

The audit messages support provides better troubleshooting and improves the cluster security by sending audit message to the configured logs.

The AFM to cloud object storage feature is enabled with the support of audit message logging for the following commands:

- **mmafmcosaccess**
- **mmafmcosconfig**
- **mmafmcosctl**
- **mmafmcoskeys**

IBM Spectrum Scale can send an audit message to the syslog and the GPFS log whenever these commands are invoked.

For more information about how to configure, see *Audit messages for cluster configuration changes* in the *IBM Spectrum Scale: Problem Determination Guide*.

Partial file or object caching for AFM to cloud object storage

When the partial file or object caching feature is enabled, an application fetches only required data from an object or a file on a cloud object storage. Therefore, network and local disk space are used more efficiently. If an application does not need to read an entire object or file, this feature must be enabled. This feature is enabled on a block boundary.

The partial caching of an object or a file is controlled by setting the **afmPrefetchThreshold** parameter value. The default value of this parameter is 0. Because of this value, an entire file is cached and all blocks of a file are fetched after any three blocks are read by the cache and the file is marked as cached. This value is useful for sequentially accessed files that are completely read such as image files. To configure this parameter, see *mmchfileset* command in *IBM Spectrum Scale: Command and Programming Reference*.

Valid values of the **afmPrefetchThreshold** parameter are in the range 1 – 100. This parameter value specifies the file size percentage that must be cached before all other the data blocks are automatically fetched into the cache. A higher value is suitable for a file that is accessed partially. When the **afmPrefetchThreshold** value is set to 100, it disables prefetching an entire file. This value caches only the data blocks that are read by an application. Also, large random-access files that do not fit in the cache are not read completely. When all data blocks are available in the cache, the file is marked as cached.

Note:

- The download or the prefetch feature fetches full objects from a cloud object storage though they are partially cached previously. Prefetch of partially cached objects pulls the entire objects to the cache.
- Failover of partially cached objects pushes only the cached data blocks to the target. Uncached blocks are filled with null bytes.
- Any writes queued on a partially cached object fetch the entire file, even if a prefetch threshold limit is set on the object.

If a write operation is queued on a file that is partially cached, the entire file is cached first, and then the write operation is queued on the file. Appending to a partially cached file does not cache the entire file. Only in the LU mode, the write inset or append on a file that is cached partially caches the entire file even if the prefetch threshold is set on the fileset.

Example

1. The *objectbucket* bucket is present on the cloud object storage with two objects as follows:

```
Name      : object100M
Date       : 2021-05-21 08:26:11 EDT
Size       : 100 MiB
ETag       : 0073370fd78dd34e5043d13ba515b5a2
Type       : file
Metadata   :
  Content-Type: application/octet-stream
```

```
Name      : object1G
Date       : 2021-05-21 08:26:11 EDT
Size       : 1000 MiB
ETag       : b7faba1ddde52a27fb925858102db50b-8
Type       : file
Metadata   :
  Content-Type: application/octet-stream
```

2. Set keys on an IBM Spectrum Scale AFM cache cluster.

```
# mmafmcosskeys objectbucket:192.168.118.121 set sdjnlSDLknsf3093mkey1
skdjfnrkfjnergergnwegwrgvklkfv12
```

3. Create an AFM to cloud object storage relationship.

```
# mmafmcossconfig fs1 objectfileset --endpoint http://192.168.118.121 --bucket objectbucket
--mode iw --object-fs --debug
afmobjfs=fs1 fileset=objectfileset
bucket=objectbucket newbucket= objectfs=yes dir=
policy= tmpdir= tmpfile= cleanup=no mode=iw
xattr=no ssl=no acls=no gcs=no vhb=
bucketName=objectbucket region= serverName=192.168.118.121 cacheFsType=http
Linkpath=/gpfs/fs1/objectfileset target=http://192.168.118.121/objectbucket
map=http://192.168.118.121 cacheHost=192.168.118.121
endpoint=192.168.118.121 ENDPOINT=--endpoint http://192.168.118.121
XOPT= -p afmParallelWriteChunkSize=0 -p afmParallelReadChunkSize=0
```

4. Verify the fileset information.

```
# mmlsfileset fs1 objectfileset --afm -L
```

A sample output is as follows:

```
Filesets in file system 'fs1':

Attributes for fileset objectfileset:
=====
Status                               Linked
Path                                 /gpfs/fs1/objectfileset
Id                                    7
Root inode                           2097155
Parent Id                             0
Created                               Fri May 21 08:31:56 2021
Comment
Inode space                           4
Maximum number of inodes              100352
Allocated inodes                     100352
Permission change flag                chmodAndSetacl
afm-associated                         Yes
Target                                http://192.168.118.121:80/objectbucket
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	0x0 (default)

Note: Prefetch threshold limit is set to 0 by default.

5. Stop the fileset.

```
# mmafmctl fs1 stop -j objectfileset
```

6. Change the prefetch threshold limit.

```
# mmchfileset fs1 objectfileset -p afmprefetchthreshold=100
```

A sample output is as follows:

```
Fileset objectfileset changed.
```

7. Start the fileset.

```
# mmafmctl fs1 start -j objectfileset
```

8. Verify the prefetch threshold limit.

```
# mmlsfileset fs1 objectfileset --afm -L
```

A sample output is as follows:

```
Filesets in file system 'fs1':

Attributes for fileset objectfileset:
=====
Status                               Linked
Path                                /gpfs/fs1/objectfileset
Id                                  7
Root inode                          2097155
Parent Id                           0
Created                             Fri May 21 08:31:56 2021
Comment
Inode space                          4
Maximum number of inodes             100352
Allocated inodes                     100352
Permission change flag               chmodAndSetacl
afm-associated                       Yes
Target                              http://192.168.118.121:80/objectbucket
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                   100
Eviction Enabled                     yes (default)
Parallel Write Chunk Size            0
IO Flags                             0x0 (default)
```

Note: The prefetch threshold is set to 100. That means only the read blocks are cached.

9. To check the objects on a cloud object storage, issue the **ls** command.

```
# ls -lash /gpfs/fs1/objectfileset/
```

A sample output is as follows:

```
total 259K
512 drwxrws--- 5 root root 4.0K May 21 08:34 .
256K drwxrwxrwx 9 root root 256K May 21 08:31 ..
0 -rwxrwxrwx 1 root root 100M May 21 2021 object100M
0 -rwxrwxrwx 1 root root 1000M May 21 2021 object1G
```

10. Read the object partially by using the **dd** command.

```
# dd if=/gpfs/fs1/objectfileset/object100M bs=4M count=10 > /dev/urandom
```

```
10+0 records in
10+0 records out
41943040 bytes (42 MB, 40 MiB) copied, 1.43027 s, 29.3 MB/s
```

```
# dd if=/gpfs/fs1/objectfileset/object1G bs=4M count=100 > /dev/urandom
```

```
100+0 records in
100+0 records out
419430400 bytes (419 MB, 400 MiB) copied, 14.129 s, 29.7 MB/s
```

11. Check the disk usage of these objects. They are the same as the data read by application or the **dd** command.

```
# du -h /gpfs/fs1/objectfileset/object1G
```

```
400M    /gpfs/fs1/objectfileset/object1G
```

```
# du -h /gpfs/fs1/objectfileset/object100M
```

```
40M     /gpfs/fs1/objectfileset/object100M
```

AFM to cloud object storage directory object support

With directory object support, you can set access control lists (ACLs) and extended attributes on empty and non-empty directories, both. These empty or non-empty directories are transferred to the home along with the files as normal operations.

AFM to cloud object storage supports directory objects. All directories with and without objects can now be synchronized to the cloud object storage. Now, you can set extended attributes and ACLs on the directories with the directory object support.

Enabling directory object support for AFM to cloud object storage

Following are the two ways to enable the directory object support for AFM to cloud object storage:

1. When you start creating a new AFM to cloud object storage relationship, use the **--directory-object** option to enable the directory object support.

```
mmafmcconfig fs1 fileset1 --endpoint http://region@endpoint --object-fs --xattr --acls
--bucket bucket1 --mode ro --directory-object
```

2. For old AFM to cloud object storage file sets that are created in old version of the IBM Spectrum Scale, use the **mmchfileset** command after you stop the file set and then start the file set.

```
node1] mmlsfileset fs1 fileset1 -L --afm
Filesets in file system 'fs1':

Attributes for fileset fileset1:
=====
Status                Linked
Path                  /gpfs/fs1/fileset1
Id                    42
Root inode            18350083
Parent Id             0
Created               Mon Apr 11 05:01:34 2022
Comment
Inode space           19
Maximum number of inodes 100352
Allocated inodes      100352
Permission change flag chmodAndSetacl
afm-associated         Yes
```

```

Permission inherit flag      inheritAclOnly
Target                      https://s3.amazonaws.com:443/fileset1
Mode                        single-writer
File Lookup Refresh Interval 120
File Open Refresh Interval  120
Dir Lookup Refresh Interval  120
Dir Open Refresh Interval    120
Async Delay                  1 (default)
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)
Parallel Write Chunk Size     0
IO Flags                     0x48080000
(afmObjectXattr,afmObjectACL,afmObjectFastReaddir)
IO Flags2                    0x0

```

```
Node1] mmafmctl fs1 stop -j fileset1
```

```
Node1] mmchfileset fs1 fileset1 -p afmObjectDirectoryObj=yes
```

```

Node1] mmlsfileset fs1 fileset1 -L --afm
Filesets in file system 'fs1':

Attributes for fileset fileset1:
=====
Status                Linked
Path                  /gpfs/fs1/fileset1
Id                    42
Root inode            18350083
Parent Id              0
Created               Mon Apr 11 05:01:34 2022
Comment
Inode space           19
Maximum number of inodes 100352
Allocated inodes      100352
Permission change flag chmodAndSetacl
afm-associated        Yes
Permission inherit flag inheritAclOnly
Target                https://s3.amazonaws.com:443/fileset1
Mode                  single-writer
File Lookup Refresh Interval 120
File Open Refresh Interval  120
Dir Lookup Refresh Interval  120
Dir Open Refresh Interval    120
Async Delay            1 (default)
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)
Parallel Write Chunk Size 0
IO Flags                0x48280000
(afmObjectXattr,afmObjectDirectoryObj,afmObjectACL,afmObjectFastReaddir)
IO Flags2               0x0
Node1] mmafmctl fs1 start -j fileset1

```

AFM to cloud object storage support for more than 2 K metadata

AFM to cloud object storage now supports setting more than 2 K metadata on the fileset objects.

The S3 protocol has limitation of 2 K total object metadata. This limitation restricts true synchronization of metadata to and from cloud object storage where objects are set with more than 2 K metadata. AFM achieves this objective by splitting the extended metadata objects and keeping them as separate files in .afm directory that is created in the cloud object storage bucket. This method is useful when fileset is used for managing large numbers of users and groups or extended attributes and access controlled lists (ACLs) that is set through distributed file systems such as NFS.

When more than 2 K metadata (xattr and ACL size exceeds 1800 bytes. As 200 bytes are reserved for other AFM metadata attributes) is set on the object, its metadata is separated in the file that is stored in

On the cloud object storage, it has object1 file in the bucket and its metadata that is stored in .afm/object1 object.

Do `ls` on bucket `fileset1` by using command-line interface similar to following sample.

```
Node1] ls aws1/pillai1/
```

```
[2022-04-29 03:19:55 EDT]    0B object1
[2022-04-29 03:23:04 EDT]    0B .afm/
```

```
Node1] ls aws1/pillai1/.afm
```

```
[2022-04-29 03:19:55 EDT] 2.7KiB object1
```

When reading or downloading an object, it reads the extended object from the .afm directory as well and populates the right data and metadata.

Note: This mechanism uses extra storage space in addition to the storage used for an actual object data on cloud object storage.

AFM to cloud object storage policy based upload for manual updates mode

AFM to cloud object storage now supports policy-based upload for manual updates (MU) mode.

A policy can be defined by system administrators and run by using the **mmafmcscctl** reconcile command that helps automatic selection and upload of the files or objects to cloud object storage buckets. Earlier, this task was done with manual intervention by using `-object-list` option from the upload command.

A policy rule is an SQL-like statement that directs **mmafmcscctl** reconcile command to upload data to the cloud object storage based on criteria defined in the policy file.

Note: Based on policy, the **mmafmcscctl** reconcile command does not upload an empty directory or a directory with objects that do not meet the policy criteria to cloud objects storage.

A policy rule specifies one or more conditions. When the conditions are true, the specific rule applies. Conditions can be specified by SQL expressions, which can include SQL functions, variables, and file attributes.

Few available file attributes are shown in the following list:

- File name or extension
- File size
- User ID and group ID
- Date and time when the file was last accessed
- Date and time when the file was last modified

Steps to policy-based upload for manual updates mode

1. According to the data created in the manual updates fileset, system administrator defines a policy. For example, upload all the objects or files with the specific names. For more information, see *Creating a policy* in the *IBM Spectrum Scale: Administration Guide*.
2. As a system administrator, you can either install this policy permanently by using **mmafmcscctl** `-add-policy` command or run it when needed, by using **mmafmcscctl** `-policy` command.
3. When the policy is installed permanently by using `-add-policy`, **mmafmcscctl** reconcile command uses this policy to run when **mafmcscctl** `Device FilesetName path reconcile` command is run without any option.
4. As a system administrator, you can remove the installed policy by using **mmafmcscctl** `Device FilesetName Path reconcile -remove-policy` command.

5. As a system administrator, you can also run a policy right away by using **mmafmcosctl** Device FilesetName Path reconcile -policy command. Here, the policy is not stored internally.
6. As a system administrator, you can view the installed policy by using --list-policy option.

Note: Reconcile command also performs cleanup of files when it is run along with the policy based uploads. Files that are deleted from manual update mode fileset are queued for deletes on the Cloud Object Storage storage and their inodes are reclaimed.

Policy creation

As a system administrator you can define a policy for variety of matching options, here is the policy that uploads all files and directories with matching name "IBM".

Important: When policy is created, make sure that the LIST option must have prerequisite names that are, "dirtyFiles" for files and "dirtyDirs" for directories.

Example

Manual updates mode fileset

```
Node1] mmlsfileset fs1 mufileset --afm -L
```

Filesets in file system 'fs1':

```
Attributes for fileset mufileset:
=====
Status                               Linked
Path                                /gpfs/fs1/mufileset
Id                                  125
Root inode                           102236163
Parent Id                             0
Created                             Wed Apr 20 03:54:03 2022
Comment
Inode space                           99
Maximum number of inodes              100352
Allocated inodes                      100352
Permission change flag                chmodAndSetacl
afm-associated                        Yes
Permission inherit flag                inheritAclOnly
Target                               https://s3.amazonaws.com:443/mufileset
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                               0x8280000
(afmObjectXattr,afmObjectDirectoryObj,afmObjectACL)
IO Flags2                             0x0
Node1]
```

Files present in Manual updates mode fileset

```
Node1] ls -l /gpfs/fs1/mufileset/
```

```
total 0
-rw-r--r-- 1 root root 0 Apr 29 08:42 extrafile1
-rw-r--r-- 1 root root 0 Apr 29 08:42 extrafile2
-rw-r--r-- 1 root root 0 Apr 29 08:42 extrafile3
-rw-r--r-- 1 root root 0 Apr 29 08:42 extrafile4
-rw-r--r-- 1 root root 0 Apr 29 08:42 extrafile5
-rw-r--r-- 1 root root 0 Apr 29 08:41 file1
-rw-r--r-- 1 root root 0 Apr 29 08:41 file2
-rw-r--r-- 1 root root 0 Apr 29 08:41 file3
-rw-r--r-- 1 root root 0 Apr 29 08:41 file4
```

```
-rw-r--r-- 1 root root 0 Apr 29 08:41 file5
-rw-r--r-- 1 root root 0 Apr 29 08:42 IBM1
-rw-r--r-- 1 root root 0 Apr 29 08:42 IBM2
-rw-r--r-- 1 root root 0 Apr 29 08:42 oneIBM
-rw-r--r-- 1 root root 0 Apr 29 08:42 twoIBM
```

```
Policy to transfer files which have IBM in their names respectively
Node1] cat policyfile1
RULE EXTERNAL LIST 'dirtyFiles'
RULE 'dirtyFilesRule' LIST 'dirtyFiles'
WHERE NAME LIKE '%IBM%' AND
PATH_NAME NOT LIKE '$filesetPath/.pconflicts/%' AND
PATH_NAME NOT LIKE '$filesetPath/.afm/%' AND
PATH_NAME NOT LIKE '$filesetPath/.ptrash/%'
```

Issue the following command at Node1]:

```
mmafmcscctl fs1 mufileset /gpfs/fs1/mufileset reconcile --policy policyfile1
```

```
Dirty file list : /var/mmfs/afm/fs1-125/recovery/policylist.data.list.dirtyFiles
  Queued      (Total)      Failed      TotalData
              (approx in Bytes)
      4          (4)          0              0
```

Object Upload successfully queued at the gateway.

Could not find any deleted files.

```
Node1] mmafmctl fs1 getstate
```

Fileset Name	Fileset Target	Cache State	Gateway
Node Queue Length	Queue numExec		
-----	-----	-----	
mufileset	https://s3.amazonaws.com:443/mufileset	Active	
c7f2n04	4 691		

File match in name as IBM are transferred to cloud objects (seen by using cloud object shell):

```
[2022-04-29 08:45:37 EDT] 0B STANDARD IBM1
[2022-04-29 08:45:37 EDT] 0B STANDARD IBM2
[2022-04-29 08:45:37 EDT] 0B STANDARD oneIBM
[2022-04-29 08:45:37 EDT] 0B STANDARD twoIBM
Node1]
```

Support of Google cloud storage platform for AFM to cloud object storage

With Google Cloud Platform support, data can be synchronized between AFM, S3 object to Google Cloud Platform storage.

AFM to cloud object storage fileset can be configured to use Google Cloud Storage (GCS) Platforms backend object storage server by using **-gcs** parameter during creation of the AFM fileset. After a fileset is created by using **-gcs** option, AFM access GCS backend storage server by using supported APIs. Here, Google cloud storage APIs are different than the traditional S3 API.

After a fileset is configured with GCS, the fileset can only be used for GCS backend server. AFM internally determines which API to use for communicating to the target cloud object storage server for all the communications.

All the modes and operations are supported by the Google Cloud Object Storage server backend.

Note:

- While you use Google Cloud Storage platform server as backend cloud object storage server, AFM is unable to create new bucket. User with an administrator privilege must create bucket at the object storage server before user creates AFM to cloud object storage fileset. This preceding limitation only exists with the GCS backend and not with other supported cloud object storage server.

- Failover of AFM to cloud object storage fileset that is configured by using `-gcs` option cannot be failover to another backend.

Symbolic links

Active File Management (AFM) to cloud object storage supports symbolic links, which are also called symlinks.

When symlinks are created on an AFM to cloud object storage cache fileset, the relationship between the symlink file and the target file is maintained and the files are uploaded to the bucket. The symlinks relationship is maintained only if they are created in an AFM to cloud object storage fileset. If the symlinks relationship is not created in the AFM to cloud object fileset, it is not maintained.

The eviction of symlinks is not supported. AFM to cloud object storage does not follow the symlink to evict an original file. The eviction might throw an error while processing a symlink file and the next file is processed.

Introduction to system health and troubleshooting

IBM Spectrum Scale comes with several functions to monitor and maintain the health of a system.

System Health

In IBM Spectrum Scale, system health monitoring is performed by the `Sysmonitor` daemon. The `Sysmonitor` daemon monitors all critical aspects of the entire cluster, where IBM Spectrum Scale is used, to ensure that potential issues are detected as soon as possible. `Sysmonitor` daemon does hundreds of checks on the relevant cluster nodes and raises RAS events. Based on these event checks, it informs the user whether anything is not working as expected and also provides guidance to solve existing problems.

You can configure IBM Spectrum Scale to raise events when certain thresholds are reached. As soon as one of the metric values exceeds or drops beyond a threshold limit, the `Sysmonitor` daemon receives an event notification from the monitoring process. The `Sysmonitor` daemon then generates a log event and updates the health status of the corresponding component. For more information about event type and health status, see *Event type and monitoring status for system health* in the *IBM Spectrum Scale: Problem Determination Guide*.

Every component has certain events that are defined for it. The `mmhealth node eventlog` command gives an overview of the happenings across all components on the local node that is sorted by time. For more information, see *System health monitoring use cases* in the *IBM Spectrum Scale: Problem Determination Guide*. The user can also create and raise custom health events in IBM Spectrum Scale. For more information, see *Creating, raising, and finding custom defined events* in the *IBM Spectrum Scale: Problem Determination Guide*.

The `mmhealth node show` command displays the results of the health monitoring of the node and its services, which run in the background. The role of a node in monitoring determines the components that need to be monitored. For many IBM Spectrum Scale components, separate categories exist in the output of the `mmhealth node show` command. For example, the typical components that are presented on a node are GPFS, network, file system, disk, and Perfmon. For a complete list of supported components, see *Monitoring the health of a node* in the *IBM Spectrum Scale: Problem Determination Guide*.

For these services, the `mmhealth node show <service>` command displays the results of the health monitoring, aggregated health state of a service, and recent active events for this service. This view also can be used to get more details about the particular health states of a service subcomponent by using the `--verbose` option. For more information about node role and functions, see *Monitoring the health of a node* in the *IBM Spectrum Scale: Problem Determination Guide*.

You can also use the `mmhealth cluster show` command to see an overview of the health monitoring for the complete cluster.

You can use the `mmhealth` command to do the following tasks:

- View the health of a node or cluster.
- View current events, and get tips for a better system configuration.
- View details of any raised event.
- Browse event history.
- Manage performance thresholds.
- Configure monitoring intervals.

For more information, see *mmhealth command* in the *IBM Spectrum Scale: Command and Programming Reference* guide.

Protocol monitoring

You can monitor system health, query events, and perform maintenance and troubleshooting tasks that are related to CES by using the **mmces** command. If a CES node is unable to export information by using the configured protocols, then its CES IP address is reassigned to another node. The reassignment is done so that when a single node goes down, the availability is not impacted. For more information, see *CES configuration issues* in the *IBM Spectrum Scale: Problem Determination Guide*.

You can use the **mmces** command to manage protocol addresses, services, node state, logging level, and balancing the load. For more information about **mmces** command, see *mmces command* in the *IBM Spectrum Scale: Command and Programming Reference* guide.

You can use the **mmprotocoltrace** command to collect trace information for debugging system problems or performance issues. For more information, see *mmprotocoltrace command* in *IBM Spectrum Scale: Command and Programming Reference* guide.

Troubleshooting

Despite all the functions that are meant to maintain your system's health, you might still face some issues with your storage clusters. To start the troubleshooting process, collect details of the issues reported in the system.

IBM Spectrum Scale provides the following options for collecting details:

- Logs
- Dumps
- Traces
- Diagnostic data collection through CLI.

Note: For more information, see *CLI commands for collecting issue details* in the *IBM Spectrum Scale: Problem Determination Guide*.

- Diagnostic data collection through GUI

For more information, see *Troubleshooting* in the *IBM Spectrum Scale: Problem Determination Guide*.

To diagnose the cause of an issue, it might be necessary to gather some extra information from the cluster. This information can then be used to determine the root cause of an issue. Collection of debugging information, such as configuration files and logs can be gathered by using the **gpfs.snap** command. This command gathers data about GPFS, operating system information, and information for each of the protocols. For more information, see *gpfs.snap command* in the *IBM Spectrum Scale: Command and Programming Reference* guide.

For more high-level analysis of an issue, you can also use the tracing feature. Tracing is logging at a high level. For more information, see *Collecting details of issues by using logs, dumps, and traces* in the *IBM Spectrum Scale: Problem Determination Guide*.

Introduction to performance monitoring

IBM Spectrum Scale provides functionality to monitor and maintain the performance of a cluster and its nodes.

IBM Spectrum Scale comes with a scalable performance monitoring solution built out of sensors and proxies, which reads the individual data, and a collector database that stores it for later use. Sensors, collectors, and proxies, along with performance metrics form the basic components of the Performance Monitoring tool. The performance metrics are counters that monitor the working of a system on which IBM Spectrum Scale is deployed. The tool can be configured to have several sensors that collect performance data from the nodes in a cluster, and scales up to many nodes. In IBM Spectrum Scale, there are more than 50 performance sensors that track more than a thousand performance metrics, and collect capacity and usage information. For more information, see *Using the performance monitoring tool* in the *IBM Spectrum Scale: Problem Determination Guide*.

The sensors first receive the data for one point in time and then parse the data into a format that is understood by the collector. The sensors then send the data directly to the collector. Queries are used by the customer or other applications to see and further use the time series data. A single collector can easily support up to 150 sensor nodes. The Performance Monitoring tool can be configured with multiple collectors to increase scalability and fault-tolerance. This latter configuration is referred to as federation. In a multi-collector federated configuration, collectors must be aware of their peers to work properly. All collectors that are part of the federation must be specified in the peer configuration option in the collector's configuration file.

You can use the **mmperfmon** command to configure the Performance Monitoring tool and its components to set up metrics, run queries, and compare node metrics. For more information, see *mmperfmon command* in the *IBM Spectrum Scale: Command and Programming Reference* guide.

The Performance Monitoring tool monitors the system on a per-component basis. A list of all supported components and metrics can be found in the *List of performance metrics* in the *IBM Spectrum Scale: Problem Determination Guide*.

A file system or disk can have several items of this kind that are called *entities* in the system. The monitoring of such a component is broken down to the monitoring of the individual file systems or disks.

All the information that is collected by the Performance Monitoring tool can be viewed by using one of the following options:

IBM Spectrum Scale GUI

A fully customizable dashboard that provides hard-wired performance charts in detail views.

mmperfmon query

A CLI command to query performance data.

Grafana bridge

An open source monitoring dashboard.

REST API to query performance data

The APIs to query and chart the performance data by user-defined dashboard software.

All the performance metrics data that is collected by the Performance Monitoring tool can be monitored by using threshold monitoring. Threshold monitoring is a service that helps the user to identify performance issues by defining threshold rules for selected performance metrics. An IBM Spectrum Scale user can set user-defined thresholds on any performance metric. For more information, see *Threshold Monitoring for system health* in the *IBM Spectrum Scale: Problem Determination Guide*.

While troubleshooting, you can also find more detailed information about the performance monitoring by viewing the perfmon tool logs. The performance monitoring tool logs can be found in the `/var/log/zimon` directory on each node that is configured for performance monitoring. For more information, see *Performance monitoring tools logs* in the *IBM Spectrum Scale: Problem Determination Guide*.

Data protection and disaster recovery in IBM Spectrum Scale

The IBM Spectrum Scale installation should be protected against data loss to ensure continuity of operation after a malfunction.

Data loss can be prevented by protecting the four types of key data:

- Cluster configuration data
- File system configuration data
- File system contents (user data, metadata, configuration)
- Protocol configuration data

Cluster configuration data is the administrative data that associates the nodes, addresses, networks, and software installation on each node. The system administrators should save the following configuration information:

- The output of the **mmlscluster** command to ensure that reconstruction of this data is possible if needed.
- CCR backup file or the **mmsdrifs** file depending on type of repository. For more information, see the *mmsdrrestore* command in *IBM Spectrum Scale: Command and Programming Reference*.
- Snapshots

File system configuration data consists of a wide variety of information involving all the file systems in the cluster. To protect this data, it is essential to use the **mmbackupconfig** command for each file system and save the output files for future use. This configuration data details which disks are associated as NSD components of which file systems, how much storage is in use, the filesets defined, the quotas defined, and other valuable configuration data that describes the file system structure. These do not include the file data in the user files. User file data is the lowest level of information and most frequently changing contents that need protection.

IBM Spectrum Scale has built in data protection processes that allows the users not only to back up and restore valuable data, but also recover data that could be potentially lost or corrupted by their own actions. For example, unintentional deletion or overwriting of a user's file.

Data backup options in IBM Spectrum Scale

This section describes the various options available in IBM Spectrum Scale to back up data.

Backing up the data

IBM Spectrum Scale creates a second copy of the data within the system as backup. These secondary copies are usually point-in-time copy of the original data, and can be created on a daily, monthly, or weekly basis. In a situation where the actual data is corrupted, the system uses the backed-up copy to restore data to a previous point in time.

You can use the **mmbackup** command to back up the user data from a GPFS file system or independent fileset to an IBM Spectrum Protect server or servers. The **mmbackup** command can be used only to back up file systems owned by the local cluster. For more information on how to back up your data, see *Backup your data* in the *IBM Spectrum Scale: Problem Determination Guide* and *mmbackup* command in the *IBM Spectrum Scale: Command and Programming Reference*.

Fileset and backup

IBM Spectrum Scale supports backup from independent filesets in addition to backup of the whole file system. For more information on fileset backup, see *Filesets and backup* in the *IBM Spectrum Scale: Administration Guide*. If you are planning to use IBM Spectrum Protect to back up IBM Spectrum Scale file systems, see [“Backup considerations for using IBM Spectrum Protect”](#) on page 270

Data restore options in IBM Spectrum Scale

This section describes the various options available in IBM Spectrum Scale to restore data.

Restoring data

The system can restore the data if it becomes corrupted due to user errors, hardware failure, and loss or even if the data gets corrupted due to bugs in other application software. For more information on how to restore system data, see *Restoring data and system configuration* in the *IBM Spectrum Scale: Problem Determination Guide*.

Data mirroring in IBM Spectrum Scale

In IBM Spectrum Scale you can copy data from one location to a secondary storage location, thus creating a mirror image of the original. This replication is called data mirroring.

In data mirroring the data is copied in real time, so the information stored in the secondary copy is always an exact replica of the data in the primary copy. Data mirroring is useful in the speedy recovery of critical data after a disaster. Data mirroring can be implemented locally or offsite at a different location. For more information on data mirroring and replication, see *Data mirroring and replication* in the *IBM Spectrum Scale: Administration Guide*.

Protecting file data using snapshots

A snapshot of an entire file system or of an independent fileset can be created to preserve the contents of the file system or the independent fileset at a single point in time.

A fileset snapshot is a snapshot of the entire inode space. Any snapshot of an independent fileset also includes any dependent filesets contained within that independent fileset. You cannot create a snapshot of a dependent fileset. For more information on creating and maintaining snapshots, see *Creating and maintaining snapshots of file systems* in the *IBM Spectrum Scale: Administration Guide*.

Snapshots can be used in environments where multiple recovery points are necessary. Ensure that the file system is mounted before restoring the snapshot. For information about mounting a file system, see *Mounting a file system* in *IBM Spectrum Scale: Administration Guide*. For more information on restoring data using snapshots, see *Restoring a file system from a snapshot* in the *IBM Spectrum Scale: Administration Guide*.

Note:

For object protocol, the best method to ensure consistency in the event that data has to be restored from a snapshot, is to configure the `cesSharedRoot` directory to be in the same file system as the object filesets.

Introduction to Scale Out Backup and Restore (SOBAR)

Scale Out Backup and Restore (SOBAR) is a data protection mechanism used specifically for disaster recovery situations.

SOBAR is used to back up and restore GPFS files that are being managed by IBM Spectrum Protect for Space Management. For more information on SOBAR, see *Scale Out Backup and Restore (SOBAR)* in the *IBM Spectrum Scale: Administration Guide*.

Commands for data protection and recovery in IBM Spectrum Scale

The following section lists the commands used for data protection and recovery in IBM Spectrum Scale.

mmbackup

Used to back up the user data from a GPFS file system or independent fileset to an IBM Spectrum Protect server or servers. For more information on **mmbackup**, see *mmbackup command* in the *IBM Spectrum Scale: Command and Programming Reference*.

mmbackupconfig

Used to collect basic file system configuration information. The configuration information backed up by this command includes block size, replication factors, storage pool layout, filesets and junction points, quota information, and a number of other file system attributes. This command forms a part of the SOBAR utilities along with the **mmrestoreconfig**, **mmimgbackup** and **mmimgrestore** commands. For more information on **mmbackupconfig**, see *mmbackupconfig command* in the *IBM Spectrum Scale: Command and Programming Reference*.

mmfsck

Used to detect and repair conditions that can cause problems in a file system. The **mmfsck** command operates in two modes: online or offline. The online mode can run while the file system is mounted. It detects and repairs lost blocks (blocks that are allocated but do not belong to any file) and corruptions in the block allocation map. The offline mode can run only when the file system is unmounted, but it can detect and fix problems that the online mode cannot. In general, you do not need to run the command in offline mode unless you are directed to by the IBM Support Center. For more information see the topic *mmfsck command* in the *IBM Spectrum Scale: Command and Programming Reference*.

mmimgbackup

Used to perform a backup of a single GPFS file system metadata image. This command forms a part of the SOBAR utilities along with the **mmbackupconfig**, **mmrestoreconfig** and **mmimgrestore** commands. Make sure that you run the **mmbackupconfig** command before you run the **mmimgbackup** command. For more information on **mmimgbackup**, see *mmimgbackup command* in the *IBM Spectrum Scale: Command and Programming Reference*.

mmimgrestore

Used to restore a single GPFS file system metadata image. This command forms a part of the SOBAR utilities along with the **mmbackupconfig**, **mmrestoreconfig** and **mmimgbackup** commands. The **mmrestoreconfig** command must be run before running the **mmimgrestore** command. For more information on **mmimgrestore**, see *mmimgrestore command* in the *IBM Spectrum Scale: Command and Programming Reference*.

mmrestoreconfig

Used to restore file system configuration information. The **mmrestoreconfig** command allows you to either query or restore, or both query and restore the output file generated by the **mmbackupconfig** command. This command forms a part of the SOBAR utilities along with the **mmbackupconfig**, **mmimgbackup** and **mmimgrestore** commands. For more information on **mmrestoreconfig**, see *mmrestoreconfig command* in the *IBM Spectrum Scale: Command and Programming Reference*.

mmrestorefs

Used to restore user data and attribute files to a file system or an independent fileset by using those of the specified snapshot. For more information on **mmrestorefs**, see *mmrestorefs command* in the *IBM Spectrum Scale: Command and Programming Reference*.

mmsdrrestore

Restores the latest GPFS system files on the specified nodes. When the CCR committed directories of all the quorum nodes have corrupted or missing files, the **--ccr-repair** parameter repairs the CCR committed directories of all the quorum nodes. For more information see *mmsdrrestore command* in the *IBM Spectrum Scale: Command and Programming Reference*.

IBM Spectrum Scale GUI

You can configure and manage various features that are available with the IBM Spectrum Scale system by using the IBM Spectrum Scale management GUI.

You can complete the following important tasks through the IBM Spectrum Scale management GUI:

- Monitoring the performance of the system based on various aspects.
- Monitoring system health.
- Creating and managing file systems.
- Creating filesets and snapshots.

- Managing Object, NFS, and SMB data exports.
- Creating administrative users and defining roles for the users.
- Creating object users and defining roles for them.
- Defining default, user, group, and fileset quotas.
- Creating and manage node classes.
- Monitoring the capacity details at various levels, such as file system, pools, filesets, users, and user groups.
- Monitoring and managing various services that are available in the system.
- Monitoring remote clusters.
- Configuring call home.
- Configuring and monitoring thresholds.
- Creating and managing user-defined node classes.
- Configuring user authentication for NFS and SMB users.

The following table provides an overview of the features that are associated with each GUI page.

<i>Table 10. Features associated with IBM Spectrum Scale GUI pages</i>	
GUI Page	Function
Home	Provides an overall summary of the IBM Spectrum Scale system configuration, status of its components, and services that are hosted on it.
Monitoring > Dashboards	Provides a dashboard to display the predefined performance charts and the customized charts that are marked as favorite charts in the Monitoring > Statistics page.
Monitoring > Events	Monitor and troubleshoot the issues that are reported in the system.
Monitoring > Tips	Monitor events of type "tips". The tip events give recommendations to the user to avoid certain issues that might occur in the future.
Monitoring > Statistics	View performance and capacity data in customizable charts.
Monitoring > Thresholds	Configure various thresholds based on the performance monitoring metrics. You can also monitor the threshold rules and the events that are associated with each rule.
Monitoring > Command Audit Log	Logs the commands that are issued and tasks that are completed by the users and administrators. These logs can also be used to troubleshoot issues that are reported in the system.
Monitoring > Event Notifications	Configure event notifications to notify administrators when events occur in the system.
Nodes	Monitor the performance and health status of the nodes that are configured in the cluster. You can also create and manage user-defined node classes from the Nodes page.
Cluster > Network	Monitor the performance and health status of various types of networks and network adapters.

Table 10. Features associated with IBM Spectrum Scale GUI pages (continued)

GUI Page	Function
Cluster > Remote Connections	Request access to the GUI node of the remote cluster to establish communication between the local and remote clusters. You can also use the Remote Connections page to grant access to the remote clusters when an access request is received.
Files > File Systems	Create, view, and manage file systems.
Files > Filesets	Create, view, and manage filesets.
Files > Snapshots	Create snapshots or snapshot rules to automate snapshot creation and retention.
Files > User Capacity	Monitor the capacity details of users and user groups.
Files > Quotas	Create and manage default, user, group, and fileset quotas at the file system level.
Files > Information Lifecycle	Create, manage, and delete policies that manage automated tiered data storage.
Files > Transparent Cloud Tiering	Provides both a summarized and attribute-wise details of the Transparent Cloud Tiering service, which is integrated with the IBM Spectrum Scale system. The Transparent Cloud Tiering page is displayed if the feature is enabled on the cluster.
Files > Active File Management	Helps to monitor the performance, health status, and configuration aspects of AFM, AFM DR, and gateway nodes. You can see the Active File Management page in the GUI that is part of the AFM cache cluster with at least one AFM-enabled fileset.
Files > File System ACL	Define ACL templates and apply ACLs on files and directories.
Files > Clustered Watch Folder	Provides the list of the clustered watch folders that are configured in the system. You can also view the details of each watch folder and the events that are raised against each record from the detailed view.
Files > Fileset tiering	Helps to run AFM policy to tier MU filesets for a specific file system.
Storage > Pools	Helps to monitor the performance, health status, and configuration aspects of all available pools in the IBM Spectrum Scale cluster.
Storage > NSDs	Helps to monitor the performance, health status, and configuration aspects of all the network shared disks (NSD) that are available in the IBM Spectrum Scale cluster.
Protocols > NFS Exports	Create and manage NFS exports. Protocols pages are displayed in the GUI only when the protocol feature is enabled on the cluster.

Table 10. Features associated with IBM Spectrum Scale GUI pages (continued)

GUI Page	Function
Protocols > SMB Shares	Create and manage SMB shares. Protocols pages are displayed in the GUI only when the protocol feature is enabled on the cluster.
Protocols > Data access service	Configure, edit and delete Data Access Service (DAS) accounts, services and exports.
Object > Accounts	Create and manage accounts and containers in the object storage. Object pages are displayed in the GUI only when the object feature is enabled on the cluster.
Object > Users	Create object users.
Object > Roles	Define roles for the object users.
Services	Monitor and manage various services that are hosted on the system.
Support > Diagnostic Data	Download diagnostic data to troubleshoot the issues that are reported in the system.
Support > Call Home	Configure call home feature that automatically notifies the IBM Support personnel about the issues that are reported in the system. You can also manually upload diagnostic data files and associate them with a PMR through the GUI.

Note: 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.

Assistance for understanding the features associated with a GUI page

The following three levels of assistance are available for the GUI users:

Hover help

A brief description of a feature that is associated with a field. It appears when you hover the mouse over the tiny question mark that is placed next to the field label. Hover help is available only for the important and complex fields.

Context-sensitive help

A detailed explanation of the features that are associated with the page. The context-sensitive help files are available in the help menu, which is placed on the header bar of the GUI page.

IBM Docs

The third-level of information helps users to find entire details of the product. You can also access the IBM Spectrum Scale documentation from the help menu, which is placed on the header bar of the GUI page.

Related concepts

[“Manually installing IBM Spectrum Scale management GUI” on page 379](#)

The management GUI provides an easy way for the users to configure, manage, and monitor the IBM Spectrum Scale system.

[“Manually upgrading the IBM Spectrum Scale management GUI” on page 529](#)

You can upgrade the IBM Spectrum Scale management GUI to the latest version to get the latest features. You can upgrade one GUI node at a time without shutting down IBM Spectrum Scale on other nodes to ensure high availability.

IBM Spectrum Scale management API

With the IBM Spectrum Scale management API, you can develop scripts to automate labor-intensive cluster management tasks. These APIs provide a useful way to integrate and use the IBM Spectrum Scale system.

The IBM Spectrum Scale management API is a REST-style interface for managing IBM Spectrum Scale cluster resources. It runs on HTTPS and uses JSON syntax to frame data inside HTTP requests and responses. For more information, see [“Functional overview” on page 150](#).

The IBM Spectrum Scale management API implementation is based on GUI stack. GUI server is managing and processing the API requests and commands. The following figure illustrates the architecture of API.

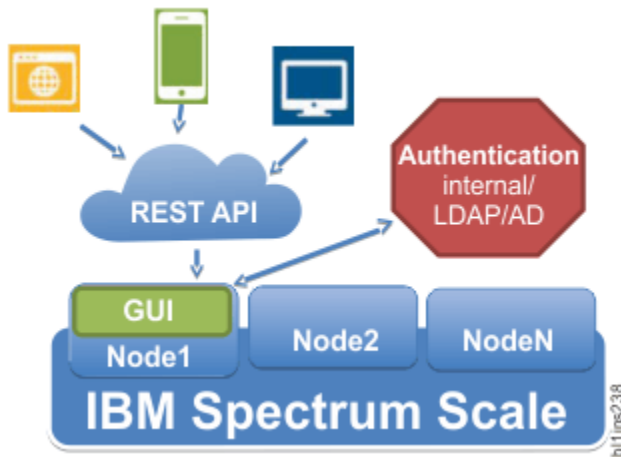


Figure 18. IBM Spectrum Scale management API architecture

The IBM Spectrum Scale management API has the following features:

- Reuses the GUI deployment's backend infrastructure, which makes introduction of new API commands easier.
- Uses the same role-based access feature that is available to authenticate and authorize the GUI users. No additional configuration is necessary for the API users. However, access to individual REST APIs might be defined at the user group-level with the help of access control lists (ACLs). Users belonging to the specific user group can access the REST APIs based on the privileges that are defined within those ACLs. Every ACL is distinct for a user group. That is, user privileges for individual REST APIs vary for different user groups.
- Makes deployment easier as the GUI installation takes care of the basic deployment.
- Fixes scalability issues and introduces new features such as, filter and field parameters and it supports paging.
- Highly scalable and can support large clusters with up to 1000 nodes.
- The APIs are driven by the same WebSphere® server and object cache that is used by the IBM Spectrum Scale GUI.

Installing and configuring IBM Spectrum Scale management API

For information about how to configure the IBM Spectrum Scale system to start using the APIs, see [Chapter 8, “Installing and configuring IBM Spectrum Scale management API,” on page 479](#).

IBM Spectrum Scale management API command reference

For information about the API endpoints, see the topic *IBM Spectrum Scale management API commands* in the *IBM Spectrum Scale: Command and Programming Reference*.

You can also access the documentation corresponding to each API endpoint from the GUI itself. The API documentation is available in the GUI at: <https://<IP address or host name of API server>:<port>/api/explorer/>. For example, <https://scalegui.ibm.com:443/api/explorer>.

The IBM Spectrum Scale management API features

The following topics describe the API features in detail.

Functional overview

The IBM Spectrum Scale management API is a REST-style API that provides interoperability between a client and server over a network. These APIs allow authenticated users to manage day-to-day storage tasks.

The following list provides the salient features of the REST-style APIs:

- Resource-based
- Stateless
- Client/server
- Cacheable
- Layered system

REST is a resource-based service system in which requests are made to the resource's Universal resources identifier (URI). The request invokes a response from the resource in the JSON format. An IBM Spectrum Scale management API *resource* is a collection of all the objects of the same type in a cluster, such as all the filesets. A *resource element* is an instance of a resource, such as a particular fileset named fileset1. In a REST API request, a resource or a resource followed by a resource element appears as the last term in the request, except for any trailing HTTPS parameters. The following example request ends with a resource, *filesets*. It indicates that this operation affects all the filesets that are part of a file system named *gpfs0*:

```
https://198.51.100.1:443/scalemgmt/v2/filesystems/gpfs0/filesets
```

In contrast, the following example ends with a resource that is followed by a resource element *filesets/fileset1*. It indicates an operation that affects this particular fileset, such as creating or deleting the fileset:

```
https://198.51.100.1:443/scalemgmt/v2/filesystems/gpfs0/filesets/fileset1
```

The kind of operations that can be performed on the resources or a resource element are directed by the HTTP methods, such as GET, POST, PUT, DELETE. In some cases, the parameters in the HTTPS request also direct the operations. The following list provides the meanings of the basic HTTP methods that are used in the requests:

- GET - Reads a specific resource or a collection of resources and provides the details as the response.
- PUT - Updates a specific resource or a collection of resources.
- DELETE - Removes or deletes a specific resource.
- POST - Creates a resource.

API requests

API requests include a URL and in some cases HTTP parameters or JSON data.

URL for an API request

The URL for an API request has the following format:

```
https://IP address or host name:port/scalemgmt/v2/resource_type/[resource_element]
```

where

IP address or host name:port

Specifies the IP address or hostname and the port of the IBM Spectrum Scale management API server, which is IBM Spectrum Scale GUI server.

/scalemgmt/v2

Specifies the path of the API server. This path is not configurable; it must be `/scalemgmt/v2`.

resource_type

Specifies a resource type. The following list specifies the supported resource types.

- ces/addresses
- ces/services
- cluster
- config
- filesets
- filesystems
- info
- jobs
- nfs/export
- nodes
- nodes/health
- nodeclasses
- quotas
- smb/shares
- snapshots
- perfmon
- /filesystems/{filesystemName}/filesets/{filesetName}/psnaps
- thresholds

Note: The full list of resource types can be accessed from the REST API explorer that is available at `https://<GUI server IP>:<port number>/api/explorer/`.

resource_element

Specifies the name of a particular resource, such as the name of a file system, a fileset, a node, or a snapshot. Optional.

As described earlier, an API request that ends with a resource type indicates a request that affects all the instances of that resource in the cluster. For example, the following HTTP request, sent with a GET method, returns information about all the nodes in the cluster:

```
'https://198.51.100.1:443/scalemgmt/v2/filesystems/nodes'
```

In contrast, an API request that ends with a resource element that follows the resource type, indicates a request that affects only the specified resource element. For example, the following HTTP request, sent with a GET method, returns information about the specified node, `node1`.

```
'https://198.51.100.1:443/scalemgmt/v2/filesystems/nodes/node1'
```

Data in API requests

Some types of API requests require one or more input parameters that follow the resource or resource element. For example, sent with a GET method, the following request returns information about all the filesets in the specified file system:

Table 11. Getting information about filesets	
Request method	GET
URL	https://198.51.100.1:443/scalemgmt/v2/filesystems/gpfs0/filesets/fs1
JSON data	None.

Also, some API requests require input parameters to be included in the body of the HTTP request in JSON format. For example, the following request creates a new snapshot in a file system. The JSON data specifies the name of the file system and the name of the new fileset:

Table 12. Creating a snapshot	
Request method	POST
URL	https://198.51.100.1:443/scalemgmt/v2/filesystems/gpfs0/snapshots
JSON data	<pre>{ "snapshotName": "snap1" }</pre>

Request headers

The following headers are needed for requests.

```
'content-type: application/json' 'accept: application/json'
```

Fields parameter

The field parameter can be used to specify the fields of the returned object by adding the query parameter "?fields" to a GET URL. For example, ?fields=field1,field2,parentField1.field1.

You can use certain keywords to get the expected results in the response. The following list provides the basic keywords that can be used with the fields parameter:

':none:'

Returns only the fields that are mandatory to uniquely identify an object. For example, filesystemName and filesetName for a fileset. Use this keyword for queries that return more than one object. It indicates that only the keys that are needed to retrieve the full object are returned.

':all:'

Returns all fields. Use this keyword in queries to a single object.

Apart from these keywords, you can also use the following options to customize the response by using the fields parameter:

- Use the field name "." to query fields that are wrapped in a parent object.

For example, use ?fields=config.iamMode to query for "iamMode" in the following object.

```
"config" : {
  "iamMode" : "off"
}
```

- Use parent name to query for all fields of a parent. For example, ?fields=config,afm includes all fields of the parent objects "config" and "afm"

- Multiple fields can be separated by using ",". For example, ?fields=config.iamMode,status.id

Examples

The following examples display the use of various keywords in the fields parameter.

1. `curl -k -u admin:admin001 -X GET --header 'accept:application/json' 'https://198.51.100.1:443/scalemgmt/v2/filesystems/gpfs0/filesets?fields=:all:'`. This GET call returns the following response:

```
{
  "status": {
    "code": "200",
    "message": "...
  },
  "paging": {
    "next": "https://localhost:443/scalemgmt/v2/filesystems/gpfs0/filesets?lastId=1001"
  },
  "filesets": [
    {
      "config": {
        "filesetName": "myFset1",
        "filesystemName": "gpfs0",
        "path": "/mnt/gpfs0/myFset1",
        "inodeSpace": "0",
        "maxNumInodes": "4096",
        "permissionChangeMode": "chmodAndSetAcl",
        "comment": "Comment1",
        "iamMode": "off",
        "oid": "158",
        "id": "5",
        "status": "Linked",
        "parentId": "1",
        "created": "2016-12-13 13.59.15",
        "isInodeSpaceOwner": "1",
        "inodeSpaceMask": "0",
        "snapId": "0",
        "rootInode": "131075"
      },
      "afm": {
        "afmTarget": "nfs://10.0.100.11/gpfs/afmHomeFs/afmHomeFileset2",
        "afmAsyncDelay": "0",
        "afmDirLookupRefreshInterval": "60",
        "afmDirOpenRefreshInterval": "60",
        "afmEnableAutoEviction": "false",
        "afmExpirationTimeout": "100",
        "afmFileLookupRefreshInterval": "30",
        "afmMode": "read-only",
        "afmNumFlushThreads": "4",
        "afmParallelReadChunkSize": "134217728",
        "afmParallelReadThreshold": "1024",
        "afmParallelWriteChunkSize": "0",
        "afmParallelWriteThreshold": "0",
        "afmPrefetchThreshold": "0",
        "afmPrimaryID": "string",
        "afmRPO": "0",
        "afmShowHomeSnapshots": "false"
      }
    }
  ]
}
```

2. GET /filesystems/gpfs0/filesets - The default keyword :none: is used in the GET request and this call returns only the key fields as shown in the following example.

```
{
  "filesets": [
    {
      "config": {
        "filesetName": "(fs1)",
        "filesystemName": "gpfs0"
      },
      "links": {
        "self": "https://198.51.100.1:8191/scalemgmt/v2/filesystems/gpfs0/filesets/fs1"
      }
    }
  ]
}
```

```

    "config": {
      "filesetName": "afmFset1",
      "filesystemName": "gpfs0"
    },
    "links": {
      "self": "https://198.51.100.1:8191/scalemgmt/v2/filesystems/gpfs0/filesets/afmFset1"
    }
  }
]
"status": {
  "code": 0,
  "message": ""
}
}

```

3. GET /filesystems/gpfs0/filesets?fields=state.rootInode,state.id - Uses the field name "." for querying fields that are wrapped in a parent object. This GET call return the following response.

```

{
  "filesets": [
    {
      "config": {
        "filesetName": "(fs1)",
        "filesystemName": "gpfs0"
      },
      "links": {
        "self": "https://198.51.100.1:8191/scalemgmt/v2/filesystems/gpfs0/filesets/fs1"
      },
      "state": {
        "id": 1,
        "rootInode": "latest",
      }
    },
    {
      "config": {
        "filesetName": "afmFset1",
        "filesystemName": "gpfs0"
      },
      "links": {
        "self": "https://198.51.100.1:8191/scalemgmt/v2/filesystems/gpfs0/filesets/afmFset1"
      },
      "state": {
        "id": 2,
        "rootInode": 50692
      }
    }
  ]
  "status": {
    "code": 0,
    "message": ""
  }
}

```

Filter parameter

Use the filter parameter to filter the retrieved objects. For example, all filesets with ID less than 2 can be retrieved by using `?filter=status.id<2`. The fields that are used in the filter parameter are always added to the result set and therefore they do not have to be specified by using the **field** parameter. Filters can be concatenated by using `"&`.

The following list provides the supported operators for different field types:

String

"=", "!=" using a regular expression as the value is supported. For example, `?filter=config.filesetName='^fs.*'`.

Numeric

"=", "!=", "<", ">". For example, `?filter=status.id<2`.

Boolean

"=", "!=" by using "true" or "false" as the value. For example, `?filter=config.isIndependent=false`.

The following examples show how to use filter parameter in a request:

```
curl -k -u admin:admin001 -X GET -H content-type:application/json
'https://198.51.100.1:443/scalemgmt/v2/filesystems/gpfs0/filesets?filter=status.rootInode>3'
{
  "filesets" : [ {
    "config" : {
      "filesetName" : "fs1",
      "filesystemName" : "gpfs0"
    },
    "status" : {
      "rootInode" : 10752
    }
  } ],
  "status" : {
    "code" : 0,
    "message" : "List of filesets for filesystem"
  }
}
```

Retrieve all file systems with encryption="yes":

```
curl -H content-type:application/json -K -u admin:admin001 -X GET
'https://198.51.100.1:443/scalemgmt/v2/filesystems?filter=encryption=yes'
(
  "filesystems" : [ (
    "encryption" : "yes"
    "filesystemName" : "objfs"
  ) ],
  "status" : {
    "code" : 200,
    "message" : "List of all file systems"
  }
)
```

API responses

API responses provide a response code from the IBM Spectrum Scale management API server and a return code and a non-mandatory return message from IBM Spectrum Scale.

Return codes and messages

A return code and an optional return message from IBM Spectrum Scale are stored in JSON format in a status structure at the end of any returned data. A return code of 0 indicates that IBM Spectrum Scale processed the request successfully:

```
"status": {
  "message": "",
  "code": 200
},
```

A response code from the IBM Spectrum Scale management API server describes the result of the API request. The following table lists the response codes.

Table 13. Response codes. A four-column table that describes the response codes.			
Description	Code	Client retry	Return code
Success	20x	None	200 - Success (GET or PUT). 201 - New resource created (POST). 202 - Request accepted (long running command). 204 - No content (DELETE).
Client error	4xx	No, the command continues to fail.	400 - Invalid request (format error in request data) 401 - Unauthorized request (wrong credentials). 403 - Resource not accessible. 404 - Resource not found (wrong URL). 405 - Method that is not supported for this resource.

Table 13. Response codes. A four-column table that describes the response codes. (continued)

Description	Code	Client retry	Return code
Server error	5xx	Yes, in most cases	500 - Internal server error, retry (system problem) 503 - Service not available (the server is busy with other requests or is down)

Data in REST API responses

IBM Spectrum Scale management API responses always include some data in JSON format. The status structure with a return code and with a return message is always present. The return message is not mandatory and might not be displayed with the code. See the status structure at the end of the following JSON example. Also, GET methods return the requested information in JSON format. The following example is from a response to a request to GET information about CES addresses. In JSON, the parenthesis [] indicate an array. The following example contains an array that is named cesaddresses that contains two elements, each of which describes one CES address.

```
{
  "cesaddresses": [
    {
      "cesNode": 1,
      "attributes": "",
      "cesAddress": "198.51.100.10",
      "cesGroup": "",
    },
    {
      "cesNode": 2,
      "attributes": "",
      "cesAddress": "198.51.100.14",
      "cesGroup": "",
    }
  ],
  "status": {
    "message": "",
    "code": 200
  }
}
```

Response headers

The same header types appear in all responses from the REST API server. The content type must be application/json. The following set of response headers is typical:

```
{
  HTTP/1.1 200 OK
  X-Frame-Options: SAMEORIGIN
  Content-Type: application/json
  Content-Language: en-US
  Content-Length: 210
  Set-Cookie:
    LtPaToken2=yyYwMkAHcegg0e74hZhsqf5iCU9r2n19QBG09nH7uPm3Mt/vpQfVEHuhwRIWKfq1fi1t8EVn6sZJx7
    +6EpVvUqxqs9PdmIXX28DzU/wwQFxFGIMA5a2AuNAFYmZ71FwqYEEhWq5tx0oQMBHQOL7AbkyR6TUq+F1wzvZnTe1
  cwu0AYmwZr6WzWdLDj8ZMJ22k95s2PmLbmNMsuSEeSbUFmc1nZdYueierBgL7QgokS90141X2gN8YSwWxx1jFzCwsed
  MsdYvhawLhYJNA3Ik0FnFVJpeopLP4EQtcSdMPXzpX+AQHn/0XQdd6iaWfHppt;
  Path=/; HttpOnly
  Set-Cookie: JSESSIONID=0000SRpFCu8e03i3WXhDyJS2qnn:8bf7532d-fb68-4d4b-90fc-4aa4e3deefd5;
  Path=/; Secure; HttpOnly
  Date: Thu, 16 Feb 2017 15:14:52 GMT
  Expires: Thu, 01 Dec 1994 16:00:00 GMT
  Cache-Control: no-cache="set-cookie, set-cookie2"
}
```

Paging

Paging happens on the response data if more than 1000 objects are returned by the query to avoid server overload. Every response object contains an object that is called "paging", which contains a link that can then be used to retrieve the next set of objects, if available.

The URL to retrieve the next page can then be found under "paging" → "next" as shown in the following example:

```
curl -k -u admin:admin001 -X GET -H content-type:application/json
'https://198.51.100.1:443/scalemgmt/v2/filesystems/gpfs0/filesets?
filter=filesetName=''myFset1.*'''
{
  "filesets" : [ {
    "config" : {
      "filesetName" : "myFset134",
      "filesystemName" : "gpfs0"
    },
    "status" : {}
  }, {
    "config" : {
      "filesetName" : "myFset133",
      "filesystemName" : "gpfs0"
    },
    "status" : {}
  } ],
  "paging": {
    "next" : "https://198.51.100.1:443/scalemgmt/v2/filesystems/gpfs0/filesets?
filter=filesetName=''myFset1.*''"
  },
  "status" : {
    "code" : 200,
    "message" : "List of filesets for filesystem"
  }
}
```

Asynchronous jobs

All POST, PUT, and DELETE requests always run asynchronously and a job object is returned in the response data. The returned job ID can then be used to retrieve the status of a job by using the GET / scalemgmt/v2/jobs/{jobId} endpoint.

The following example shows how to get information about the job 12345.

Request URL:

```
curl -k -u admin:admin001 -X GET --header 'accept:application/json'
'https://198.51.100.1:443/scalemgmt/v2/jobs/12345'
```

Response data:

```
{
  "status": {
    "code": "200",
    "message": "... "
  },
  "paging": {
    "next": "https://localhost:443/scalemgmt/v2/filesystems/gpfs0/filesets?lastId=1001"
  },
  "jobs": [
    {
      "result": {
        "commands": "['mmcrfileset gpfs0 restfs1001', ...]",
        "progress": "['(2/3) Linking fileset']",
        "exitCode": "0",
        "stderr": "['EFSSG0740C There are not enough resources available to create
a new independent file set.', ...]",
        "stdout": "['EFSSG4172I The file set {0} must be independent.', ...]"
      },
      "request": {
        "type": "GET",
        "url": "/scalemgmt/v2/filesystems/gpfs0/filesets",
        "data": "{\n\"config\":{\n\"filesetName\":\n\"restfs1001\",\n\"owner\":\n\"root\",\n\"path\":\n\"/mnt/gpfs0/rest1001\",\n\"permissions\":\n\"555\"}\n}"
      },
      "jobId": "12345",
    }
  ]
}
```

```

    "submitted": "2016-11-14 10.35.56",
    "completed": "2016-11-14 10.35.56",
    "status": "COMPLETED"
  }
]
}

```

Note: In the JSON data that is returned, the return code indicates whether the command is successful. The response code 200 indicates that the command successfully retrieved the information. Error code 400 represents an invalid request and 500 represents internal server error.

The following endpoints are available to monitor or delete a job:

- GET /jobs: Get details of asynchronous jobs.
- GET /jobs/{jobId}: Get details of a specific asynchronous job.
- DELETE /jobs/{jobId}: Cancel an asynchronous job that is running.

Accessing the IBM Spectrum Scale REST API endpoint details through Swagger and API explorer

You can access the details of the API endpoints through the following three options:

- Swagger editor.
- API explorer option that is available on the GUI node.
- IBM Spectrum Scale Documentation.

The Swagger option is widely used because you can view the descriptions, run the API endpoints with the required parameters, and generate client code.

Accessing API details by using swagger

The Swagger editor uses the API description file, which can either be in YAML or JSON format. Such a description file describes the API formally in a machine-readable format with all the attributes, operations and endpoints. You can get the API description from the GUI node.

The following steps help you to access the API details and try out the endpoints by using the Swagger editor:

1. Retrieve the description file for the IBM Spectrum Scale management API from the following location: `https://[host name or IP address of the GUI server]:443/api/docs?root=/scalemgmt/v2/`. The following screen appears.

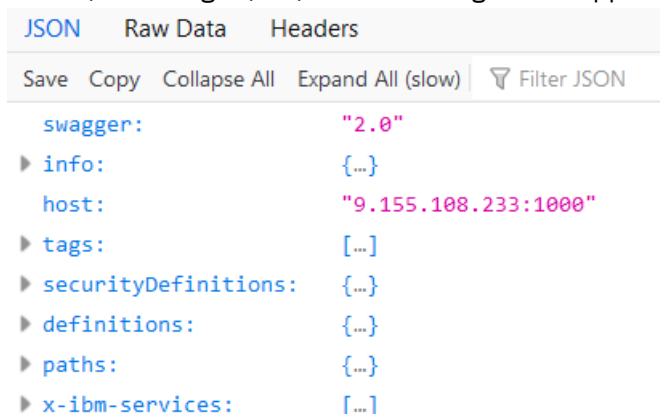


Figure 19. Option to download API description files in JSON format from the GUI node

2. Click **Save** and save the description file in the JSON format.

Note: If you are using Chrome or Microsoft Edge, you must right-click the browser page and select **Save as** from the menu that appears to save your file in the JSON format.

3. Go to <http://editor.swagger.io/> to open the Swagger editor in the web browser.
4. In the **File** menu, select **Import file** and then browse and select the JSON file that you downloaded.

You can now see the content of the description file on the left side and the generated documentation on the right side as shown in the following figure:

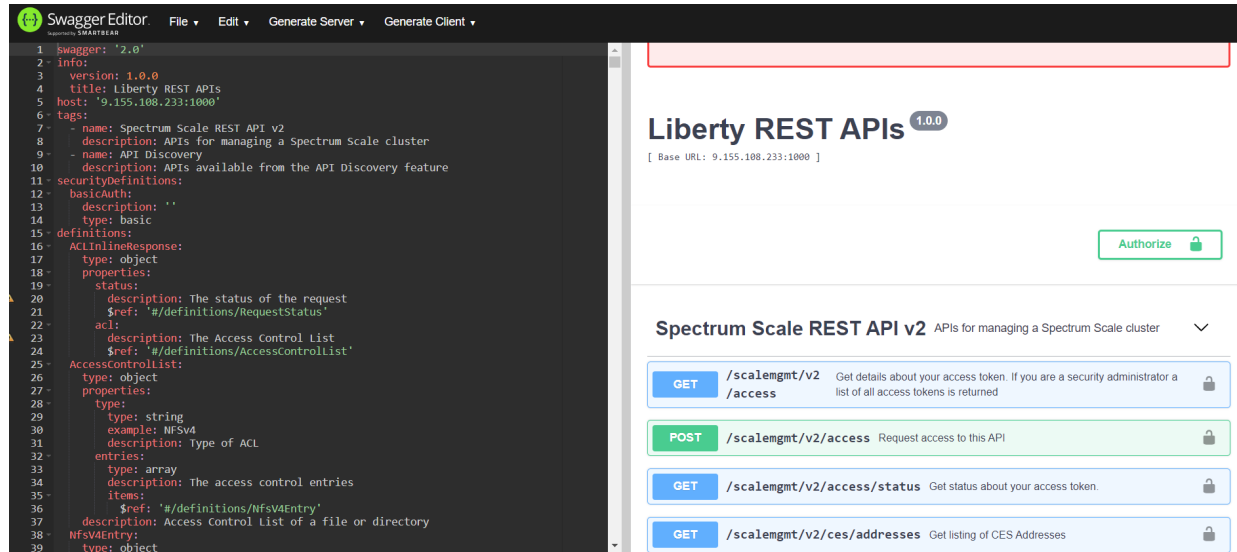


Figure 20. REST API documentation opened in Swagger editor

The left side of the view provides the code. The right side of the view displays all the endpoints with their input and output parameters. It also provides option to try out the code by using Curl. The following topics provide details about the request parameters, response, and status of the request:

- “API requests” on page 151
- “API responses” on page 155

The following steps explain how to work with the endpoints in the Swagger editor:

1. Open Swagger editor and import the description file as explained before. You can see the content of the description file on the left side and the generated documentation on the right side.
2. Click **Authorize** and enter your user credentials that you use to access the GUI node.
3. Click the endpoint that you want to explore or try out, from the list of endpoints. The endpoint details appear. Each endpoint provides the details such as the list of request parameters, option to try out the parameters, and response data that includes the requested information and endpoint execution status. The following figure shows options that are available in the GET /scalemgmt/v2/ces/services/{service} endpoint.

GET

/scalemgmt/v2/ces/services/{service}

Get detailed information about a CES Service

🔒

Returns detailed information for a CES Service

Parameters

Cancel

Name	Description
service * required string (path)	Service name to query <div>smb</div>

Execute

Responses

Response content type application/json

Code	Description
200	successful operation <div> <div>Example Value</div> <div>Model</div> <div> <pre>{ "status": { "code": 200, "message": "...", }, "cesservices": { "protocolStates": [{</pre> </div> </div>

Figure 21. Example for options that are available in the Swagger editor for each endpoint

4. Type the name of service for which you need the details, in the Parameters section. In this example, SMB is selected as the target service.
5. Click **Execute** to view the details of the specified service.

The **Responses** section displays the following details:

- Curl command that is run in the background. For example, `curl -X GET "http://198.51.100.12:1000/scalemgmt/v2/ces/services/smb" -H "accept: application/json" -H "authorization: Basic YWRtaW46YWRtaW4wMDE="`
- Request URL. In this case, it is `http://198.51.100.12:1000/scalemgmt/v2/ces/services/smb`
- Response data: The response data includes the status of the request and the requested data in a GET request.

Generating client code from Swagger

To generate the client code from the Swagger editor, click **Generate Client** and select the format in which you need to generate the client code. You can use it to develop the client application that uses the IBM Spectrum Scale API endpoints.

Accessing API details from GUI node by using the API explorer

To display a swagger-like documentation, you can access the following page in your browser:

```
https://[guiHost]:443/api/explorer/
```

Where, [guiHost] is the host name or IP address of the IBM Spectrum Scale GUI server.

The options that are available to view the endpoint details or try it out, are the same as the ones that are available in the Swagger editor. When you use the API explorer that is available on the GUI node, you do not need to separately import the JSON file that contains the API endpoint descriptions.

Accessing API documentation through the IBM Spectrum Scale Documentation

To access the details of each endpoint from the IBM Spectrum Scale Documentation, see *IBM Spectrum Scale management API endpoints* in *IBM Spectrum Scale: Command and Programming Reference* guide.

List of IBM Spectrum Scale management API commands

The following table lists the currently supported operations for each type of resource and resource element. The first and second columns list the resource type and the operation. The third column lists the resource and resource element terms that appear in the HTTPS request. The braces {} are a notation for indicating a resource element in text and do not appear in the actual HTTPS request. The fourth column lists the HTTP method that is called to do the operation. A footnote indicates a request that requires HTTP parameters.

Note: You can access the documentation corresponding to each API command from the GUI node.

The API documentation is available in the GUI at: `https://<IP address or host name of GUI server>:<port>/api/explorer/`. For example, `https://scalegui.ibm.com:443/api/explorer`.

Table 14. Operations supported for resources and resource elements in API endpoints			
Resource type	Operation	Resource/{element}	Method
CES addresses	Get information about CES addresses.	/ces/addresses	GET
	Get information about a CES address.	/ces/addresses/{cesAddress}	GET
CES services	Get information about CES services.	ces/services	GET
	Get information about a CES service.	/ces/services/{service}	GET
Cluster	Get information about the current cluster.	/cluster	GET

Table 14. Operations supported for resources and resource elements in API endpoints (continued)

Resource type	Operation	Resource/{element}	Method
GPFS snap	List the GPFS snap files containing the diagnostic data that is collected by the gpfs.snap command.	/diagnostic/snap	GET
	Create a GPFS snap file.	/diagnostic/snap	POST
	Download the specific snap file that comprises the diagnostic data to troubleshoot issues in the system.	/diagnostic/snap/{snapPath}	GET
	Upload a snap file to a PMR.	/diagnostic/snap/{snapPath}/pmr/{pmrID}	PUT
	Delete the snap file that contains the diagnostic data that is collected by the gpfs.snap command.	diagnostic/snap/{snapPath}	DELETE
CLI audit log	Gets details of the CLI commands that are issued on the cluster.	/cliauditlog	GET
Config	Get the current configuration attributes.	/config	GET
File systems	Get information about file systems in the cluster.	/filesystems	GET
	Get information about a file system.	/filesystems/{filesystemName}	GET
	Resume a file system.	filesystems/{filesystemName}/resume	PUT
	Suspend a file system.	/filesystems/{filesystemName}/suspend	PUT
	Unmount a file system.	/filesystems/{filesystemName}/unmount	PUT
ACLs	Get access control list of a file or directory in a file system.	/filesystems/{filesystemName}/acl/{path}	GET
	Set access control list for a file or directory in a file system.	/filesystems/{filesystemName}/acl/{path}	PUT
	Get the list of all REST API access control lists (ACL).	/access/acls	GET
	Get the list of the REST API access control lists (ACL) defined for a user group.	/access/acls/{userGroup}	GET
	Delete all the REST API access control lists (ACL) that is defined for a user group.	/access/acls/{userGroup}	DELETE
	Delete a specified REST API access control lists (ACL) entry that is defined for a user group.	/access/acls/{userGroup}/entry/{entryID}	DELETE
	Copy a REST API ACL entry from an existing group and overwrite the entries that are defined for the specified group.	/access/acls/{userGroup}	POST
	Add a REST API access control list (ACL) entry for a user group.	/access/acls/{userGroup}	PUT

Table 14. Operations supported for resources and resource elements in API endpoints (continued)

Resource type	Operation	Resource/{element}	Method
File audit logging	Enable or disable file audit logging for a file system.	/filesystems/{filesystemName}/audit	PUT
Storage Pool	Get information about a particular storage pool.	/filesystems/{filesystemName}/pools/{poolName}	GET
	Get the list of storage pools in a specific file system.	/filesystems/{filesystemName}/pools	GET
Disks	Get details about disks that are part of a file system.	/filesystems/{filesystemName}/disks	GET
	Get details about a disk that is part of a file system.	/filesystems/{filesystemName}/disks/{diskName}	GET
Filesets	Get information about filesets in a file system.	/filesystems/{filesystemName}/filesets	GET
	Get information about a fileset.	/filesystems/{filesystemName}/filesets/{filesetName}	GET
	Create a fileset.	/filesystems/{filesystemName}/filesets	POST
	Modify a fileset.	/filesystems/{filesystemName}/filesets/{filesetName}	PUT
	Delete a fileset.	/filesystems/{filesystemName}/filesets/{filesetName}	DELETE
	Link an existing fileset.	/filesystems/{filesystemName}/filesets/{filesetName}/link	POST
	Unlink a fileset.	/filesystems/{filesystemName}/filesets/{filesetName}/link	DELETE
	Control AFM for a fileset.	/filesystems/{filesystemName}/filesets/{filesetName}/afmctl	POST
Directory	Creates a directory on a GPFS file system.	/filesystems/{filesystemName}/directory/{path}	POST
	Remove a directory from a GPFS file system.	/filesystems/{filesystemName}/directory/{path}	DELETE
	Copy a directory on a file system.	/filesystems/{filesystemName}/directoryCopy/{sourcePath}	PUT
	Create a directory on a fileset.	/filesystems/{filesystemName}/filesets/{filesetName}/directory/{path}	POST
	Remove a directory from a fileset.	/filesystems/{filesystemName}/filesets/{filesetName}/directory/{path}	DELETE
	Copy a directory on a fileset.	/filesystems/{filesystemName}/filesets/{filesetName}/directoryCopy/{sourcePath}	PUT

Table 14. Operations supported for resources and resource elements in API endpoints (continued)

Resource type	Operation	Resource/{element}	Method
Quotas	Get information about quota defined at the fileset level.	/filesystems/{filesystemName}/filesets/{filesetName}/quotas	GET
	Set quota limits at the fileset level.	/filesystems/{filesystemName}/filesets/{filesetName}/quotas	POST
	Get information about quota defined at the file system level.	/filesystems/{filesystemName}/quotas	GET
	Set quota limits at the file system level.	/filesystems/{filesystemName}/quotas	POST
	List quota defaults in a cluster.	/filesystems/{filesystemName}/filesets/{filesetName}/quotadefaults	GET
	Set the quota defaults USR, GRP, or FILESET for a file system.	/filesystems/{filesystemName}/filesets/{filesetName}/quotadefaults	POST
	Enable/disable the quota defaults for a fileset.	/filesystems/{filesystemName}/filesets/{filesetName}/quotadefaults	PUT
	List quota defaults in the cluster.	/filesystems/{filesystemName}/quotagracedefaults	GET
	Set the default quota for user, group, and fileset of a file system.	/filesystems/{filesystemName}/quotagracedefaults	POST
	Enable or disable the quota management for a file system.	/filesystems/{filesystemName}/quotamanagement	PUT
Cloud Object Store	Create a Cloud Object Store fileset.	/filesystems/{filesystemName}/filesets/cos	POST
	Create a Cloud Object Store-related directory in a fileset.	/filesystems/{filesystemName}/filesets/{filesetName}/cos/directory	POST
	Download files from Cloud Object Store.	/filesystems/{filesystemName}/filesets/{filesetName}/cos/download	POST
	Evict files from object store.	/filesystems/{filesystemName}/filesets/{filesetName}/cos/evict	POST
	Upload files to object store.	/filesystems/{filesystemName}/filesets/{filesetName}/cos/upload	POST
Owner	Get details of the owner of the file or directory in a file system.	/filesystems/{filesystemName}/owner/{path}	GET
	Set owner of the file or directory in a file system.	/filesystems/{filesystemName}/owner/{path}	PUT

Table 14. Operations supported for resources and resource elements in API endpoints (continued)

Resource type	Operation	Resource/{element}	Method
Policies	Lists the policies that are applied on the file system.	/filesystems/{filesystemName}/policies	GET
	Sets a policy for a file system.	/filesystems/{filesystemName}/policies	PUT
Information	Get information about the IBM Spectrum Scale management API.	/info	GET
Jobs	Get details about asynchronous jobs.	/jobs	GET
	Get details about an asynchronous job.	/jobs/{jobID}	GET
	Cancel a specific asynchronous job.	/jobs/{jobId}	DELETE
NFS exports	Get information about NFS exports configured in the system.	/nfs/exports	GET
	Get information about an NFS export configured in the system.	/nfs/exports/{exportPath}	GET
	Create an NFS export.	/nfs/exports	POST
	Modify an existing NFS export.	/nfs/exports/{exportPath}	PUT
	Delete an NFS export.	/nfs/exports/{exportPath}	DELETE
Nodes	Get information about nodes in the cluster.	/nodes	GET
	Get information about a node.	/nodes/{nodeName}	GET
	Add one or more nodes in the cluster.	/nodes	POST
	Delete one or more nodes from the cluster.	/nodes/{name}	DELETE
	Change node designation for a node or node class.	/nodes/{name}	PUT
Node mapping	List all AFM node mappings.	/nodes/afm/mapping	GET
	Create AFM node mapping.	/nodes/afm/mapping	POST
	Delete existing mapping.	/nodes/afm/mapping/{mappingName}	DELETE
	Gets the details of a specific map.	/nodes/afm/mapping/{mappingName}	GET
	Change an existing map.	/nodes/afm/mapping/{mappingName}	PUT
Node health	Get details about system health events that are reported on a node.	/nodes/{name}/health/events	GET
	Get details about system health states of the events that are reported in a node.	/nodes/{name}/health/states	GET

Table 14. Operations supported for resources and resource elements in API endpoints (continued)

Resource type	Operation	Resource/{element}	Method
Node class	Get details of the node classes.	/nodeclasses	GET
	Create a node class.	/nodeclasses	POST
	Delete a specific node class.	/nodeclasses/{nodeclassName}	DELETE
	Get details of a specific node class.	/nodeclasses/{nodeclassName}	GET
	Make changed to a specific node class.	/nodeclasses/{nodeclassName}	PUT
Symlink	Remove a symlink from a GPFS file system.	/filesystems/{filesystemName}/symlink/{path}	DELETE
	Create a symlink for a path from a GPFS file system.	/filesystems/{filesystemName}/symlink/{linkPath}	POST
Services	Gets the details of the services that are configured on a node.	/nodes/{name}/services	GET
	Gets the details of a specific service that is configured on a node.	/nodes/{name}/services/{serviceName}	GET
	Starts or stops a service on a node or node class.	/nodes/{name}/services/{serviceName}	PUT
NSDs	Gets information about NSDs that are part of the system.	/nsds	GET
	Gets information about an NSD.	/nsds/{nsdName}	GET

Table 14. Operations supported for resources and resource elements in API endpoints (continued)

Resource type	Operation	Resource/{element}	Method
Snapshots	Get information about fileset snapshots.	/filesystems/{filesystemName}/ filesets/{filesetName}/ snapshots	GET
	Get information about a fileset snapshot.	/filesystems/{filesystemName}/ filesets/{filesetName}/ snapshots/{snapshotName}	GET
	Get information about file system snapshots.	/filesystems/{filesystemName}/ snapshots	GET
	Get information about a file system snapshot.	/filesystems/{filesystemName}/ snapshots/{snapshotName}	GET
	Create a fileset snapshot.	/filesystems/{filesystemName}/ filesets/{filesetName}/ snapshots	POST
	Create a file system snapshot.	/filesystems/{filesystemName}/ snapshots	POST
	Delete a fileset snapshot.	/filesystems/{filesystemName}/ filesets/{filesetName}/ snapshots	DELETE
	Delete a file system snapshot.	/filesystems/{filesystemName}/ snapshots	DELETE
	Copy a file system snapshot.	/filesystems/{filesystemName}/ filesets/{filesetName}/ snapshotCopy/{snapshotName}	PUT
	Copy a directory from a path relative to a snapshot of the file system.	/filesystems/{filesystemName}/ filesets/{filesetName}/ snapshotCopy/{snapshotName}/ path/{source Path}	PUT
AFM peer snapshots	Create a peer snapshot.	/filesystems/{filesystemName}/ filesets/{filesetName}/psnaps	POST
	Delete a peer snapshot.	/filesystems/{filesystemName}/ filesets/{filesetName}/psnaps/ {snapshotName}	DELETE
AFM	List AFM state in a file system.	/filesystems/ {filesystemName}/afm/state	GET

Table 14. Operations supported for resources and resource elements in API endpoints (continued)

Resource type	Operation	Resource/{element}	Method
SMB shares	Get information about SMB shares.	smb/shares	GET
	Get information about an SMB share.	/smb/shares/{shareName}	GET
	Create an SMB share.	/smb/shares	POST
	Modify an existing SMB share.	/smb/shares/{shareName}	PUT
	Delete an SMB share.	/smb/shares/{shareName}	DELETE
	Delete complete access control list of an SMB share.	/smb/shares/{shareName}/acl	DELETE
	Get access control list of share.	/scalemgmt/v2/smb/shares/{shareName}/acl	GET
	Delete an entry from the access control list of a SMB share.	/smb/shares/{shareName}/acl/{name}	DELETE
	Get access control list entry of a specific user/group/system of share.	/smb/shares/{shareName}/acl/{name}	GET
	Adds an entry to the access control list of an SMB share.	/smb/shares/{shareName}/acl/{name}	PUT
Performance monitoring	Get performance data.	/perfmon/data	GET
	Gets the sensor configuration.	/perfmon/sensor	GET
	Gets details about a specific sensor configuration.	/perfmon/sensor/{sensorName}	GET
	Modifies the sensor configuration.	/perfmon/sensor/{sensorName}	PUT
	Run a custom query for performance data.	/perfmon/stats	GET
Thresholds	Get list of threshold rules.	/thresholds	GET
	Create a threshold rule.	/thresholds	POST
	Delete an existing threshold rule.	/thresholds/{name}	DELETE
	Get details of a threshold rule.	thresholds/{name}	GET
For more information about the commands, see <i>REST API commands</i> in the <i>IBM Spectrum Scale: Command and Programming Reference</i> guide.			

Cloud services

Transparent cloud tiering is a feature of IBM Spectrum Scale that provides a native cloud storage tier.

Cloud services have the following two components:

- Transparent cloud tiering
- Cloud data sharing

Transparent cloud tiering allows data center administrators to free up IBM Spectrum Scale storage capacity, by moving out cooler data to the cloud storage, reducing capital and operational expenditures. Tiering can also be used to archive an extra copy of your data by using pre-migration, a function that copies data rather than moving it. The Transparent cloud tiering feature leverages the existing ILM policy query language semantics available in IBM Spectrum Scale, and administrators can define policies to

tier data to cloud storage. On an IBM Spectrum Scale cluster with multiple storage tiers configured, this external cloud storage can be used as the cooler storage tier to store infrequently accessed data from a cool storage pool. *For performance reasons, it is recommended not to move any active or hot data to this external storage pool, as it drives excessive data traffic on the Transparent cloud tiering which in turn can cause delays, leading to problems like application timeouts.* Copying hot data to the cloud by using the pre-migration function is acceptable when the file is not likely to be updated or deleted soon.

Cloud data sharing is an IBM Spectrum Scale Cloud services that allows a way to set up sharing between IBM Spectrum Scale and various types of object storage, including IBM Cloud Object Storage. Furthermore, for export of data to object storage the service can be invoked from an ILM policy to allow for periodic sharing of data based on when the policy is run. For import, you can specify a list of files that you can use to move object storage (by using your IBM Cloud data sharing service) into the IBM Spectrum Scale cluster.

This is useful when data needs to be distributed across multiple domains or when large amounts of data need to be transferred.

You can associate your Cloud services file systems and file sets with a Cloud services node group. Up to four node groups are supported in a cluster to allow for scaling (of the solution). For each file system or file set, you can associate up to two cloud storage tiers. For example, you could define an IBM Cloud Object Storage service locally to act as your cool tier. You could use IBM's Public Cloud Object Storage (on IBM Cloud) for your cold tier. The following diagram is an example of a Cloud services configuration that is on a cluster:

Figure 22 on page 169 illustrates these features.

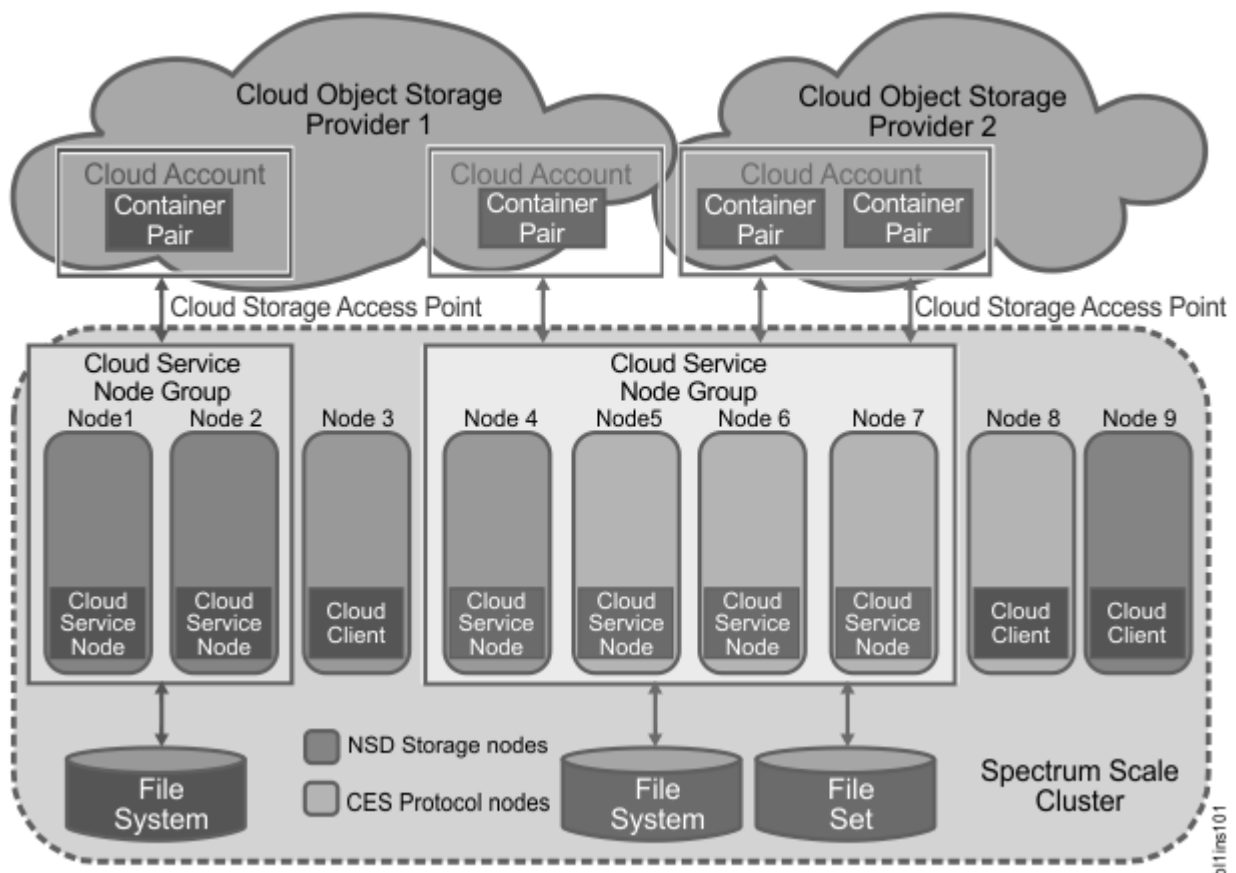


Figure 22. Transparent cloud tiering and Cloud data sharing features

Note:

- For performance reasons, the median file size to be migrated to the cloud tier must be greater than 1 MB. Migration is supported for file size less than one MB, but performance is slower because of the overhead that is associated with small files.
- For Transparent cloud tiering, data on the Cloud Object Storage is opaque and cannot be accessed directly by applications. All I/O operations must happen through your IBM Spectrum Scale system.
- Transparent cloud tiering works with IBM Spectrum Scale on multi-site stretch clusters to allow continued access to cloud storage data even if failure of an entire site.
- For applications that require tiering of data that scales beyond 100 million files per file system, you can create extra containers on cloud storage to hold your files. For applications that require more I/O bandwidth than a single maximum node group (four nodes) is capable of, you can create extra node groups.

Note: You must create a new container when the existing container has approximately 100,000,000 files.

Unsupported use cases

Transparent cloud tiering does not support the following use cases:

- Using Transparent cloud tiering to migrate or recall hot (active) data.
- Using Transparent cloud tiering as a backup mechanism.
- Using Transparent cloud tiering to restore data in disaster recovery scenarios.

Note: IBM Spectrum Protect cloud-container storage pools can be used to back up data to cloud storage providers.

Note: To enable Cloud services nodes, you must first enable the Transparent cloud tiering feature. This feature provides a new level of storage tiering capability to IBM Spectrum Scale customers. Contact your IBM Client Technical Specialist (or send an email to <mailto:scale@us.ibm.com>) to review your use case of the Transparent cloud tiering feature and to obtain the instructions to enable the feature in your environment.

How Transparent cloud tiering works

This topic describes how you can use Transparent cloud tiering (which tiers files to object storage) feature in the IBM Spectrum Scale cluster.

Transparent cloud tiering is an IBM Spectrum Scale feature that allows the IBM Spectrum Scale cluster to tier files to object storage. The Transparent cloud tiering service stores a file as two objects out in the cloud in two separate containers. These containers are also called vaults or buckets depending on which cloud storage vendor you use. One object holds the file's data and the other object holds the file's metadata. The metadata contains items such as ACLs, extended attributes, and other information that allows Transparent cloud tiering to do a full restore of the file from the cloud tier.

When a file is migrated to the cloud, the metadata of the file is retained in the IBM Spectrum Scale file system. The only reason for copying the metadata to the cloud is to manage scenarios where a file is destroyed on the cluster for some reason (such as accidental deletion) and the file needs to be restored from the cloud.

For the usual case, only the file's data is recalled from the cloud. You can tier the files by migrating the data to the cloud. This means that the data is no longer on the IBM Spectrum Scale cluster and is stored only on the cloud. Such files are considered to be in the non-resident state. Alternatively, you can pre-migrate the data to the object storage so that the data is retained in the IBM Spectrum Scale cluster, but is also copied out to object storage so that the data is on both tiers. Such files are considered to be in the co-resident state.

Transparent cloud tiering stores information about files that are migrated to the cloud tier in a database called *cloud directory*. A separate database or cloud directory is kept for every Transparent cloud tiering container pair (the pair of containers that hold the file data and metadata). This database contains a list and versions of all files that are migrated to the cloud. The metadata contains items such as ACLs, extended attributes, and other information that allows Transparent cloud tiering to do a full restore of the file from the cloud tier in the future.

Because some applications access the front end of the file frequently, there is an option by specifying "thumbnail-size" to choose how much of the front end of a file is to be kept in the file system when a file is migrated out to the object storage. Some applications such as Windows Explorer and Linux GNOME only access a very small amount of data on the front end as a part of directory services and other applications such as media streamers might want to cache hundreds of megabytes. By specifying the size it is possible to efficiently accommodate both such applications.

Transparent cloud tiering has maintenance activities that remove data from the cloud for files that are deleted or reversioned. Backup and reconciliation services are performed to deal with disaster recovery and other unusual error clean-up cases. You can change the times and frequencies of these activities.

Additionally, if a file is migrated and only the metadata is changed later, a subsequent migration copies only the metadata to the cloud. This metadata references the original data object.

Note: Transparent cloud tiering data is migrated to the cloud in a way that is not designed for direct use in the cloud. If data must be consumed in the cloud, you must consider the Cloud data sharing service.

Data migration and configuration activities happen through the defined Cloud services nodes. You can allow transparent recalls to be handled directly (and thus more efficiently and with better performance) by client nodes by setting up this option when installing the client.

How Cloud data sharing works

This topic describes how the Cloud data sharing feature works in the IBM Spectrum Scale cluster.

Cloud data sharing allows you to import data from object storage into your IBM Spectrum Scale file system using the **import** command. You can also export data to the cloud for use in the cloud by using the **export** command. Optionally, there is a manifest that can be built that allows those sharing to very quickly ascertain what files are being shared.

Cloud data sharing is different from Transparent cloud tiering because it is meant as a means of sharing and distributing data, whereas with Transparent cloud tiering the data that is migrated to object storage is concealed so it can be consumed only by recalling the data back into the IBM Spectrum Scale system.

Note: There is no implicit consistency link that is established by the export or import of a file. The import or export is just a transfer of a file. Once the transfer is complete, if the original file is changed at one place it has no bearing on its copies. And if a copy of a file is changed or deleted it has no bearing on the original.

How exporting file system data to cloud storage works

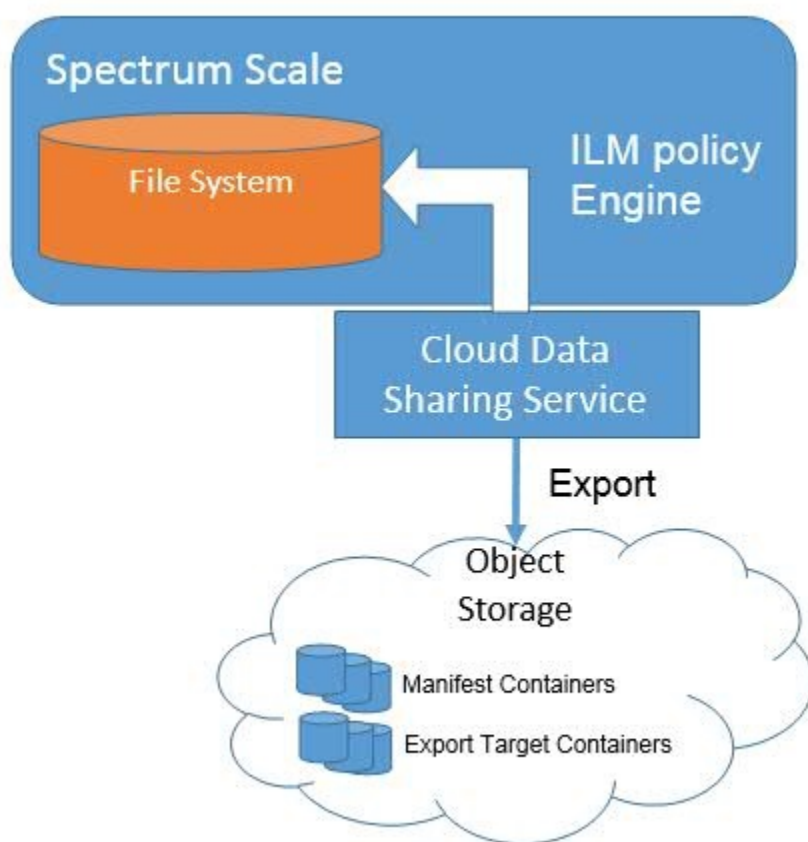


Figure 23. Exporting file system data to a cloud storage tier

You can use the Cloud data sharing service to export data to the cloud. Data can be exported manually, but typically it is done leveraging the ILM Policy Manager that comes with IBM Spectrum Scale. When data transfer requests are generated, they are distributed evenly between all nodes in the Cloud services node group providing the Cloud data sharing service.

Also, optionally as a part of the export request, a list of what was sent is also put in a manifest file that can be used to track what was exported. Export is very flexible because for each command the target container can be specified. Any container accessible by the given cloud account can be used. For each **export** command, you can also specify the target manifest file. The manifest file is not automatically stored in the cloud storage but rather remains stored within the Cloud data sharing service. The manifest is exported on demand to the desired target location.

How importing object storage data into the Spectrum Scale file system works

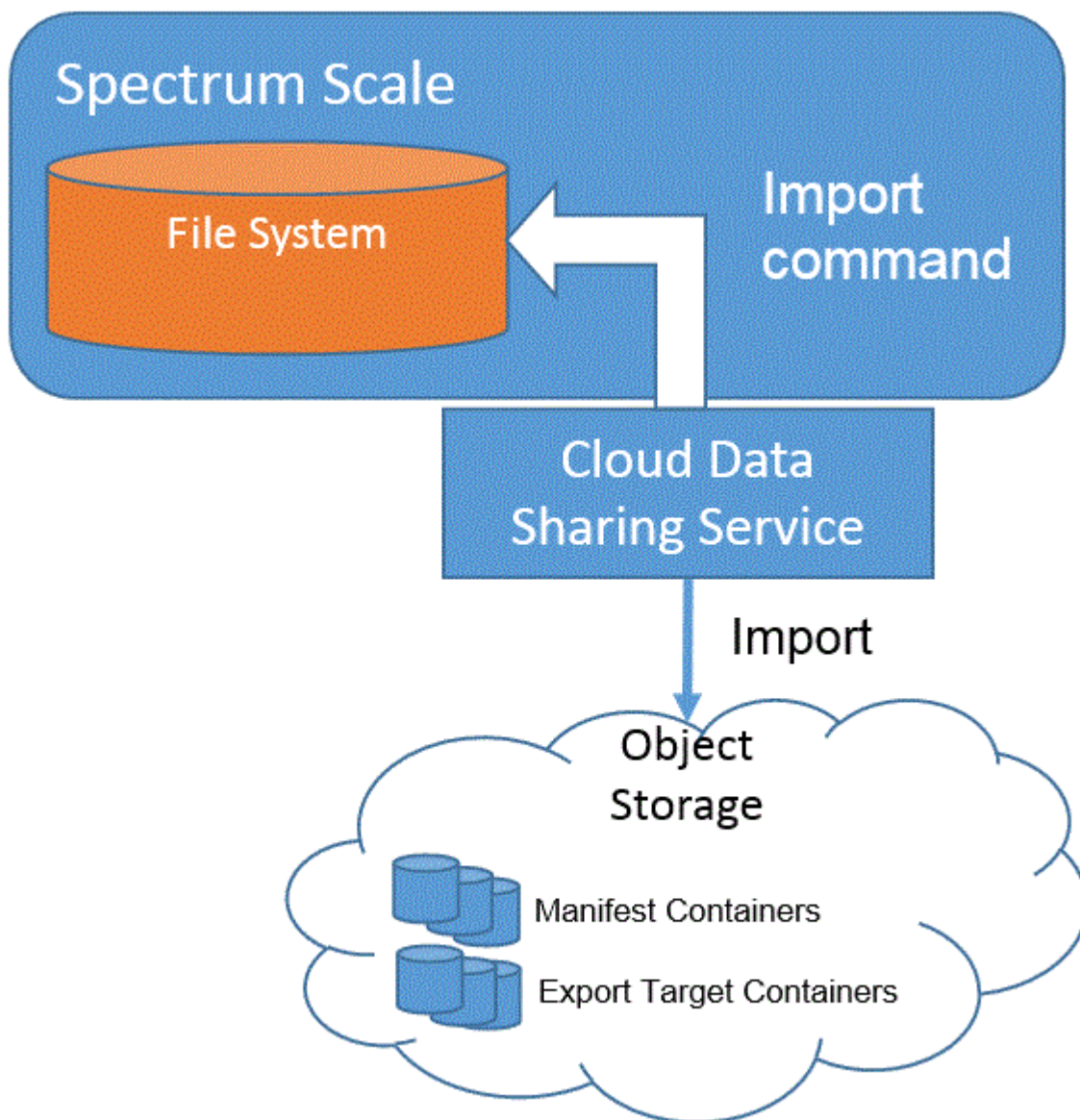


Figure 24. Importing object storage data into the file system

You can import data directly by providing a list of objects to be imported in the command itself or by providing a pointer to a manifest that contains the list of objects to be imported. The files are imported and placed in a target directory in the file system. Since the files are not yet present in the file system, there is no way to use the policy engine to initially import the files. You can also import the stub of the file and then use the policy engine to look at the file stub information that you use to import the desired data subset. You can delete the stubs that you are not interested in.

Note: The import of the stub only works if the data on the object storage that is being imported was originally created using the IBM Spectrum Scale export service so that the stub has the appropriate format. (Native object storage stubs are not mapped or normalized as of yet).

How to be aware of and exploit what data is shared

A manifest might be kept which lists information on objects that are in cloud storage available for sharing. When the Cloud data sharing service exports files to the object storage, it can be directed to add entries on what it exported to a manifest. Other applications that generate cloud data can also generate a manifest. The location of where the manifest is stored and when it is exported to be looked at is the decision of the application – it is not automatic. A typical time to export the manifest to object storage

is in the same cron job as the policy invocation immediately following the policy execution completion so that it represents all the files that are exported by that policy run.

The `mmcloudmanifest` manifest utility program is available that can read a manifest and provide a comma-separated value (CSV) output stream of all the files in the manifest, or a subset of those files as provided by some simple filtering. This manifest utility is written in Python and can run separately from the cloud data sharing service most anywhere python can run. The utility is also built into cloud data sharing and can be used to get input for import operations.

Currently, there is no built-in way to do asynchronous notification or to have a database that tracks all updates at a scale larger than is reasonable with manifest files, but it is possible to add notification in the script where the export policy is invoked. Also, manifests can be readily consumed by a database (for example the Elastic Search, Logstash, Kibana ELK Stack) since the data structure is very simple. On larger deployments, such an approach with asynchronous notification that is built into the scripts (and uses a database) is worth consideration.

How a native cloud storage application can prepare a set of objects for sharing

There is a simple way for native cloud storage applications or other generators of cloud object data to generate a manifest. The manifest utility can build the manifest from a list of objects that it is passed.

How Write Once Read Many (WORM) storage works

This topic describes how you can use Write Once Read Many (WORM) solutions to take advantage of functions that are offered by IBM Spectrum Scale, Transparent cloud tiering, and IBM Cloud Object Storage.

WORM (Write Once-Read Many) storage solutions leverage IBM Spectrum Scale immutability, Transparent cloud tiering, and IBM Cloud Object Storage locked vaults. Essentially, you can create the immutable file sets on the IBM Spectrum Scale file system, and you can set files as immutable either through IBM Spectrum Scale commands or through POSIX interface. You can also set a retention period on the immutable files.

A locked vault on the IBM Cloud Object Storage cannot be deleted by the IBM Cloud Object Storage administrator, and its Access Control Lists (ACLs) cannot be changed. Additionally, you cannot rename it or enable the proxy settings. RSA private key and private certificate are used to create and access the locked vaults. Transparent cloud tiering is configured with the RSA private key and private certificate to create locked vaults. Once configured, you can use configured private keys and certificates to use REST APIs against the Accesser nodes.

You can perform migrate and recall operation on immutable files. Since the immutable files cannot be deleted, the data on the IBM Cloud Object Storage locked vaults is only deleted according to locked vault deletion policy, which provides a WORM solution for compliance archival.

The following diagram provides an overview of the feature:

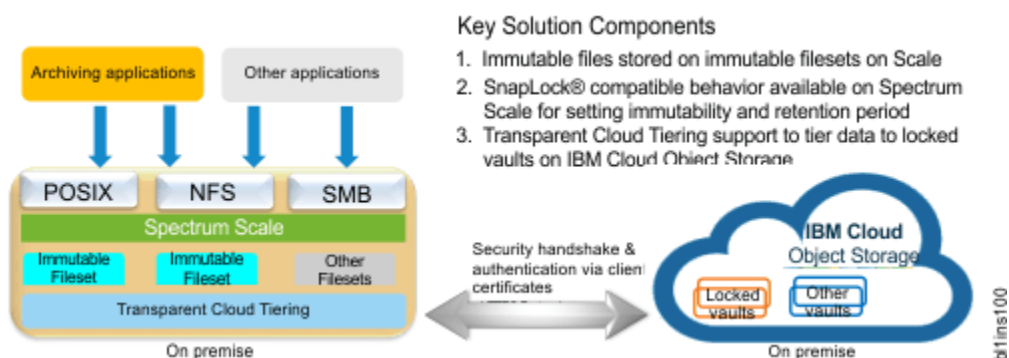


Figure 25. WORM storage overview

Supported cloud providers

This topic describes the cloud providers that the IBM Spectrum Scale supports.

The Cloud services use the following cloud providers:

- IBM Cloud Object Storage (version 3.7.3.2, 3.8.2.19, 3.9, and 3.13.4.40)
- IBM Cloud Object Storage on IBM SoftLayer® and IBM Bluemix®.
- OpenStack Swift with Swift version 2.13 and below
- Amazon Web Services S3
- Swift3 on OpenStack Swift version 2.13, 1.1 and above

Note: OpenStack Swift is supported only with Keystone version 2 only. You need to install Apache libcloud 1.3.0 for mmcloudmanifest to function. Run **pip install apache-libcloud==1.3.0** to install libcloud version 1.3.0.

- Microsoft Azure object storage service

Note: Cloud services move data to a cool tier as a block 'blob' only. Moving data to other tiers and other blob types is not supported.

Interoperability of Transparent cloud tiering with other IBM Spectrum Scale features

This topic describes the interoperability of Transparent cloud tiering with other IBM Spectrum Scale features.

IBM Spectrum Archive (LTFS) and IBM Spectrum Protect

Running IBM Spectrum Archive and Transparent cloud tiering on the same file system is not supported.

However, both IBM Spectrum Protect and Transparent cloud tiering can co-exist on the same systems if they are configured with different file systems.

It is advised not to enable the Data Management API (DMAPI) on the file system that is managed by the Transparent cloud tiering.

AFM

Running Transparent cloud tiering service on AFM Gateway nodes is not supported.

Data from AFM or AFM DR filesets cannot be accessed by Transparent cloud tiering.

Using Transparent cloud tiering on AFM home is not supported.

Multi-clusters

Multi-cluster support is limited to remotely mounted clients.

File Placement Optimizer (FPO)

Transparent cloud tiering is supported on FPO cluster. Recalled files might not have optimal placement.

IBM Spectrum Scale Object

Transparent cloud tiering can be configured on Object file sets.

Snapshots

Transparent cloud tiering should not be used to migrate/recall snapshots. Space that is contained in snapshots is not freed if a file is migrated to cloud object storage. Data tiered to the cloud is not designed for proper behavior relative to a snapshot. It is not recommended to take a snapshot on a file set or filesystem that contains data tiered by Transparent cloud tiering service.

Sparse files

Transparent cloud tiering can be used to migrate/recall sparse files. However, sparseness is not maintained and on recall, full blocks are allocated.

Encryption

Transparent cloud tiering can be used with Scale file system level encryption feature (available in the Advanced and Data Management Editions). However, all the data that is migrated to the cloud object storage is migrated with the key that is configured for Transparent cloud tiering. Essentially, when data is read from file system, the files get decrypted, and encrypted again at user space by Transparent cloud tiering and pushed into the cloud storage.

Compression

Transparent cloud tiering can be used along with Scale file system level compression capability. Essentially, when data is read from the file system, the files are uncompressed, Transparent cloud tiering push uncompressed, but by default encrypted, files onto the cloud storage.

CES (protocol services)

Transparent cloud tiering can co-exist with active or inactive NFS, SMB, or Object services on the CES nodes.

IBM Spectrum Protect

For the file systems that are managed by IBM Spectrum Protect system, ensure that hot data is backed up to IBM Spectrum Protect by using the **mmbackup** command, and as the data gets cooler, migrate them to the cloud storage tier. This ensures that the **mmbackup** command has already backed up the cooler files that are migrated to the cloud.

Elastic Storage Server (ESS)

Transparent cloud tiering cannot be deployed directly on ESS nodes. However, it can be deployed on X86 protocol nodes that can be attached to ESS. On a Power Linux cluster, a few X86 protocol nodes can be added and Transparent cloud tiering can be configured on those nodes. Remotely mounted client support is also helpful when running with ESS configurations since multi-cluster configurations with ESS are very common.

Mixed-node cluster configuration

Transparent cloud tiering service runs on x86, Power LE Linux nodes, and IBM Z nodes. Transparent cloud tiering does not run on Windows or Power BE Linux nodes. No mixed-node cluster support with Windows. Both x86 Linux and Power Linux nodes can initiate migrations/recalls of data, and these nodes can initiate a transparent recall on file access.

Transparent cloud tiering is supported for use only in IBM Spectrum Scale clusters and on any associated remotely mounted clients, with Intel x86 Linux and Power LE Linux nodes. Use of Windows, PowerPC® Big Endian, Linux on Z or AIX nodes that access file systems, where Transparent cloud tiering is used (either within the cluster or via a remotely mounted client), is untested and unsupported. Inclusion of such nodes is allowed only if there is no possibility that they will ever access Transparent cloud tiering; for example, such nodes are tested and supported for PowerPC Big Endian nodes in Elastic Storage Server (ESS) BE & LE servers, where no user applications are supported on Elastic Storage Server (ESS).

SELinux

Supported

SOBAR

SOBAR backup and restore support is available.

IPV6 Support

Not supported

IBM Spectrum Scale Stretch Clusters

This service can be used in conjunction with stretch clusters. Cloud services node classes can be set up all on one site or, most usually, can be split across sites to allow for continued access to cloud data even through failure of an entire site.

installation toolkit

Not supported.

Linux on Power8 Little Endian

Supported.

Note: For information on known limitations, see the *Known Limitations for Transparent cloud tiering* topic in the *IBM Spectrum Scale: Administration Guide*.

Interoperability of Cloud data sharing with other IBM Spectrum Scale features

This topic describes the interoperability of Cloud data sharing with other IBM Spectrum Scale features.

IBM Spectrum Archive (LTFS) and IBM Spectrum Protect

- Running IBM Spectrum Archive and Cloud data sharing on the same file system is not supported.
- However, both IBM Spectrum Protect and Cloud data sharing can co-exist on the same systems if they are configured with different file systems.
- It is advised not to enable the Data Management API (DMAPI) on the file system that is managed by the Cloud data sharing.

AFM

- Data from AFM or AFM DR filesets might be accessed by Cloud data sharing.
- Using Cloud data sharing on AFM home is supported.

Multi-clusters

Running Cloud data sharing on multiple clusters to the same or different cloud accounts is supported.

File Placement Optimizer (FPO)

Cloud data sharing is supported on FPO cluster. Transferred files might not have an optimal placement.

IBM Spectrum Scale Object

- Cloud data sharing works only with Unified File Object based IBM Spectrum Scale object storage at this time.
- Running Cloud data sharing in the same file system with another data set is supported.

Snapshots

Cloud data sharing can be used with the snapshot function to export snapshot files to the cloud since it is simply doing the file replication.

Sparse files

Cloud data sharing can be used to import or export sparse files. However, sparseness is not maintained and on re-import, full blocks are allocated.

Encryption

Cloud data sharing can be used with the IBM Spectrum Scale file system level encryption feature (available in the Advanced Edition or IBM Spectrum Scale Erasure Code Edition). However, all the data that is exported to the cloud object storage is unencrypted or if cloud data sharing encryption is enabled it is exported with the encryption key configured for Cloud services. Essentially, when data is read from file system, the files get decrypted, and may be encrypted again at user space by Cloud data sharing and pushed into the cloud storage.

Compression

Cloud data sharing can be used along with IBM Spectrum Scale file system level compression capability. Essentially, when data is read from the file system, the files will get uncompressed, Cloud data sharing will push uncompressed file onto the cloud storage.

CES (protocol services)

Cloud data sharing can co-exist with active or inactive NFS, SMB, or Object services on the CES nodes.

Spectrum Scale stretched cluster

Spectrum Scale stretched cluster and Cloud data sharing on the same file system is supported.

Spectrum Protect (TSM)

IBM Spectrum Scale file systems backed up using Spectrum Protect may leverage Cloud data sharing.

Elastic Storage Server (ESS)

Cloud data sharing cannot be deployed directly on ESS nodes. However, it can be deployed on x86 Protocol (CES) nodes that can be attached to ESS that effectively allows sharing of ESS data.

Mixed-node cluster configuration

- Cloud data sharing service runs only on x86 and Power Linux nodes. Cloud data sharing is not supported to be deployed on Power Linux or Windows. However, it can co-exist in a Linux cluster that contains both x86 and Power nodes, as long as the Power nodes are not running the Cloud services.
- No mixed-node cluster support with Windows. Only x86 Linux nodes can initiate migrations/recalls of data and only x86 Linux nodes can initiate a transparent recall on file access.

SELinux

Supported

SOBAR

Running SOBAR and Cloud data sharing on the same file system is supported.

IPv6 Support

Not supported

Linux on Power8 Little Endian

Supported

installation toolkit

Not supported.

Note: For information on known limitations, see the *Known Limitations for Cloud services* topic in the *IBM Spectrum Scale: Administration Guide*.

File audit logging

File audit logging captures file operations on a file system and logs them to a retention enabled fileset.

Each file operation is generated as a local event on the node that serves the file operation. These events are produced to the audit fileset. These events are called lightweight events. Lightweight events occur at the file system level and capture all accesses to a monitored file system from protocol exports to even root access that occurs directly on nodes. For more information, see [“Producers in file audit logging” on page 179](#). The most common file operations such as open, close, rename, unlink, create, remove directory, extended attribute change, ACL change, and GPFS attribute change are the events that are captured. Events are created in a highly parseable JSON formatted string as they are written to the designated fileset. For each file system enabled for file audit logging, a fileset is designated for the audit logs. This fileset keeps the logs currently being written to in append-only mode and, as it rotates to a new log file, compresses the old log file and makes it immutable for the retention period. Configurable options for file audit logging filesets include their name, whether or not the audit fileset should be IAM mode compliant, and the retention period in days. An entire file system can be audited, a subset of filesets within the file system can be audited, or the file system can be audited while skipping events in selected filesets. File audit logging is integrated into the system health infrastructure, so alerts are generated for the producer and state changes of the producer.

Note: IBM Spectrum Scale file audit logging is the preferred method for logging file system activities.

For more information about file audit logging, see *Monitoring file audit logging* in *IBM Spectrum Scale: Administration Guide*.

Producers in file audit logging

Producers create lightweight events and publish them to the audit fileset.

Lightweight events are generated when file operations are served locally on a node in the data path. These lightweight events are generated as node local events that are then published to the audit fileset. They reside in the GPFS daemon and are activated and configured by IBM Spectrum Scale policy code.

The file audit logging fileset

The audited events are logged in an independent file audit logging fileset.

The user can also supply a previously created fileset if it is already in IAM mode noncompliant (or IAM mode compliant if the **--compliant** flag is given). Using IAM mode noncompliant or compliant is a requirement so that audit logs are not altered accidentally. This requirement also allows a retention period to be set on the files within the audit fileset. By default, the file audit logging fileset is IAM mode noncompliant, so the root user can change the expiration date on files and then delete them if needed to free up space. If file audit logging is enabled with the optional **--compliant** flag, then not even the root user can change the expiration date on files within the audit fileset. In that case, no files can be deleted before the expiration date. This previously created fileset must also be linked directly from the root mount point: `<FS_Mount_Point>/<Fileset_Name>`. The user specifies this fileset during enablement by using the **--log-fileset** option on the **mmaudit** command.

File audit logging records for a file system are arranged in the following manner:

```
<FS_Mount_Point>/<Fileset_Name>/Topic/Year/Month/Day
```

- **FS_Mount_Point**: The default mount point of the device being audited is designated to hold the file audit logging destination fileset.
- **Fileset_Name**: The fileset name indicates the file audit logging destination fileset. If no other fileset name is given, then the default is `.audit_log`.
- **Topic**: The topic that is associated with the file system that is being audited. It consists of the following structure:

```
<Device_Minor Number>_<Cluster_ID>_<Generation_Number>_audit
```

- **Year/Month/Day**: Details when the individual file that is being written was first created.

Note: The **mmbackupconfig** command does not save the file audit logging enabled state of a file system. However, the IAM noncompliant or IAM-compliant mode of the file audit logging fileset is saved. Therefore, when you restore the configuration information with the **mmrestoreconfig** command, file audit logging must be reenabled on the file system with the **mmaudit** command. You need to specify **--log-fileset** and any other enable flags if you are not using the default values. For more information, see *IAM modes and their effects on file operations on immutable files* in the *IBM Spectrum Scale: Administration Guide*.

File audit logging records

Files that hold the JSON formatted event records.

Files in the file audit logging fileset are named in the following format:

```
auditLogFile_<IO_Node_Name >_YYYY-MM-DD_HH:MM:SS
```

When a file audit logging file is created, it is put in append only mode and an expiration date is set for it. The expiration date corresponds to the number of days after the current date given in the retention option of the **mmaudit** command. The default retention period is 365 days, which means that a given file audit logging file is set to expire one year from when it is created.

When auditing events are generated by file system operations, they are appended to the files in the file audit logging fileset. After approximately 5,000,000 entries have been made in a file or the file size exceeds 400MB, whichever comes first, it is made immutable, GPFS native compression is enabled on it,

and a new file with the current date and time is created for the next batch of audit logs. Therefore, new files are created at a rate that corresponds to the file system activity of the device that is being audited.

Important: Since files in the file audit logging fileset are created in the directory based on their creation date, there might be audit records for several days within the same file if there is not much file system activity.

Furthermore, since the events are written to these files in batches, the file audit logging event information might be placed in a previous day's or next day's directory. In order to improve usability, there are soft links that are created in the audit fileset topic subdirectory that link each current audit record file. These links are named in the following format: `auditLogFile.latest_<IO_Node_Name>`.

For more information, see [“JSON attributes in file audit logging” on page 181](#).

File audit logging events

Use this information to learn more about which I/O operations result in the ten events for file audit logging.

Table 15. File audit logging events		
Event name	Description	Examples
ACCESS_DENIED	A user was denied access to operate on a file.	<code>open()</code> with <code>O_WRONLY</code> where user has no write permission.
ACLCHANGE*	A file's or directory's ACL permissions were modified.	<code>mmputacl</code>, <code>chown</code>, <code>chgrp</code>, <code>chmod</code>
CLOSE	A file was closed.	<code>close()</code> , <code>cp</code> , <code>touch</code> , <code>echo</code> , policy MIGRATE rule.
CREATE*	A file or directory was created.	<code>open(create flag)</code> , <code>vi</code> , <code>ln</code> , <code>dd</code> , <code>mkdir</code>
GPFSATTRCHANGE*	A file's or directory's IBM Spectrum Scale attributes were changed.	<code>mmchattr -i -e --indefinite-retention</code>
OPEN	A file or directory was opened for reading, writing, or creation.	<code>open()</code> , <code>mmlsattr</code> , <code>cat</code> , <code>cksum</code> , <code>ls</code> (only for directories), policy LIST rule
RENAME*	A file or directory was renamed.	<code>rename()</code> , <code>mv</code>
RMDIR*	A directory was removed.	<code>rmdir()</code> , <code>rm</code> , <code>rmdir</code>
UNLINK*	A file or directory was unlinked from its parent directory. When the linkcount = 0, the file is deleted.	<code>unlink()</code> , <code>rm hardlink/softlink</code>
XATTRCHANGE*	A file's or directory's extended attributes were changed.	<code>mmchattr --set-attr --delete-attr</code>

Note: The * shows that these events are not applicable to a file system mounted as read-only.

For more information, see *JSON reporting issues in file audit logging* in *IBM Spectrum Scale: Problem Determination Guide*.

JSON attributes in file audit logging

Use this information to learn more about the JSON attributes that are associated with the 10 events in file audit logging.

LWE_JSON

The version of the record.

path

The path name of the file that is involved in the event.

oldPath

The previous path name of the file during the **RENAME** event. For all other events, it is not displayed.

clusterName

The name of the cluster where the event took place.

nodeName

The name of the node where the event took place.

nfsClientIp

The IP address of the remote client that is involved in the event.

fsName

The name of the file system that is involved in the event.

event

This is one of the following events: **OPEN**, **CREATE**, **CLOSE**, **RENAME**, **UNLINK**, **XATTRCHANGE**, **ACLCHANGE**, **RMDIR**, **GPFSATTRCHANGE**, or **ACCESS_DENIED**.

inode

The inode number of the file that is involved in the event.

linkCount

The Unix link count of the file that is involved in the event.

openFlags

The open flags that are specified during the event. For example:

```
fcntl.h ( O_RDONLY,O_WRONLY,O_RDWR, O_CREAT, ...)
```

For example:

```
"openFlags": "32962" = 0x80C2 = o100302 translates to ( O_RDWR | O_CREAT | O_EXCL | O_LARGEFILE)
```

poolName

The pool name where the file resides.

fileSize

The current size of the file in bytes.

ownerUserId

The owner ID of the file that is involved in the event.

ownerGroupId

The group ID of the file that is involved in the event.

atime

The time in UTC format of the last access of the file that is involved in the event.

ctime

The time in UTC format of the last status change of the file that is involved in the event.

mtime

The time in UTC format of the last modification to the file that is involved in the event.

eventTime

The time in UTC format of the event.

clientUserId

The user ID of the process that is involved in the event.

clientGroupId

The group ID of the process that is involved in the event.

accessMode

The access type for which the operation was denied for the ACCESS_DENIED event.

processId

The process ID that is involved in the event.

bytesRead

The bytes read from a file.

bytesWritten

The bytes written to a file.

minReadOffset

The starting position of bytes read from a file.

maxReadOffset

The ending position of bytes read from a file.

minWriteOffset

The starting position of bytes written to a file.

maxWriteOffset

The ending position of bytes written to a file.

permissions

The permissions on the file that is involved in the event.

acls

The access control lists that are involved in the event.

xattrs

The extended attributes that are involved in the event.

subEvent

The type of IBM Spectrum Scale attribute change. Only applies to the immutability and appendOnly flags.

The following table describes the JSON attributes that are provided for the 10 events in file audit logging:

Table 16. JSON attributes in file audit logging										
Attribute	OPEN	CREATE	CLOSE	RENAME			UNLINK	RMDIR		ACCESS_DENIED
LWE_JSON	X	X	X	X	X	X	X	X	X	X
path	X	X	X	X	X	X	X	X	X	X
oldPath				X						
clusterName	X	X	X	X	X	X	X	X	X	X
nodeName	X	X	X	X	X	X	X	X	X	X
nfsClientIp	x ¹	x ¹	x ¹	x ¹	x ^{1,2}	x ¹	x ¹	x ¹		x ¹
fsName	X	X	X	X	X	X	X	X	X	X
event	X	X	X	X	X	X	X	X	X	X
inode	X	X	X	X	X	X	X	X	X	X
linkCount	X	X	X	X	X	X	X	X	X	
openFlags	X	0	X	0	0	0	0	0	0	X
poolName	X	X	X	X	X	X	X	X	X	X
fileSize	X	X	X	X	X	X	X	X	X	X
ownerUserId	X	X	X	X	X	X	X	X	X	X

Table 16. JSON attributes in file audit logging (continued)										
Attribute	OPEN	CREATE	CLOSE	RENAME			UNLINK	RMDIR		ACCESS_DENIED
ownerGroupId	X	X	X	X	X	X	X	X	X	X
atime	X	X	X	X	X	X	X	X	X	X
ctime	X	X	X	X	X	X	X	X	X	X
mtime	X	X	X	X	X	X	X	X	X	X
eventTime	X	X	X	X	X	X	X	X	X	X
clientId	X	X	X	X	X	X	X	X	X	X
clientGroupId	X	X	X	X	X	X	X	X	X	X
processId	X	X	X	X	X	X	X	X	0	X
bytesRead	0	Null	X ⁵	Null	Null	0	Null	Null	Null	Null
bytesWritten	0	Null	X ⁵	Null	Null	0	Null	Null	Null	Null
minReadOffset	MAX INT	Null	X ⁵	Null	Null	MAX INT	Null	Null	Null	Null
maxReadOffset	0	Null	X ⁵	Null	Null	0	Null	Null	Null	Null
minWriteOffset	MAX INT	Null	X ⁵	Null	Null	MAX INT	Null	Null	Null	Null
maxWriteOffset	0	Null	X ⁵	Null	Null	0	Null	Null	Null	Null
permissions	X	X	X	X	X	X	X	X	X	X
acls	Null	Null	Null	Null	Null	X	Null	Null	Null	Null
xattrs	Null	Null	Null	Null	X ³	Null	Null	Null	Null	Null
subEvent	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	APPENDONLY IMMUTABILITY	NONE
accessMode	Null	Null	Null	Null	Null	Null	Null	Null	Null	X

Note: In the above table, 0, Null, or MAX INT represents that the attribute is not applicable for that particular event.

For more information about some of the issues that might occur with the events and when they might occur, see in the *IBM Spectrum Scale: Problem Determination Guide*.

Note:

1. The **nfsClientId** attribute is provided for NFS clients that use Ganesha. The value is NULL for kernel NFS versions and SMB.
2. The **nfsClientId** attribute is populated for an **XATTRCHANGE** event when SELinux is enabled and a **CREATE** event is generated via NFS.
3. The **xattrs** attribute only shows the **xattr** that was changed.
4. The best effort is made to provide both **path** and **nfsClientId** attributes for files accessed via NFS, but it is not guaranteed.
5. Results might be inaccurate with mmap IO, mmap IO via SMB, or IO via NFS.

Remotely mounted file systems in file audit logging

In IBM Spectrum Scale, events are recorded for file systems that are remotely mounted when file audit logging is enabled on the **owningCluster**.

Events are only recorded on nodes that meet the file audit logging prerequisites. For more information, see [“Requirements, limitations, and support for file audit logging” on page 489](#). File audit logging commands do not work on file systems that are not owned by the local cluster. File audit logging can

be enabled for both local and remotely mounted file systems at the same time with audit records that are stored by the **owningCluster** of the file system. To determine whether a file system is under audit from an **accessingCluster**, submit the **mmfsfs FS --file-audit-log** command.

Clustered watch folder

You can use the clustered watch folder to monitor file systems, filesets, and inode spaces for file accesses.

With the clustered watch folder feature, it is possible to watch file operations across clusters by using a centralized tool that has scalability and resiliency built-in. An entire file system, fileset, or inode space can be watched. Clustered watch folder captures file system activities at the IBM Spectrum Scale file system level, generates event notifications for those activities, and streams the notifications to topics in an external sink that you can manage.

Producers in clustered watch folder

Producers generate file access event notifications.

Producers are located in the IBM Spectrum Scale daemon. For a file system with an enabled clustered watch, a producer exists in the daemon on each node that mounts the file system. The producer generates events for a clustered watch if the producer is on the IBM Spectrum Scale cluster that owns the file system and the clusters with the file system remotely mounted. In addition, both clusters must meet the minimum IBM Spectrum Scale level versions. The producer code requires two packages to be installed. For more information, see [“Manually installing clustered watch folder”](#) on page 492.

Interaction between clustered watch folder and the external Kafka sink

Clustered watch folder supports sending watch events to an external Kafka sink.

To send watch events to an external Kafka sink, a minimum of two attributes must be present when you enable a clustered watch:

- A list of accessible broker addresses with ports to access the external Kafka queue.
- The topic name on the external Kafka queue where the clustered watch publishes events.

In addition to the two required attributes, authentication or authorization can also be specified. If authentication or authorization is not given when you enable a clustered watch, it is assumed that it is not needed. The following types of authentication or authorization are supported:

- **NONE**: This is the default. It can also be specified by excluding any type of authentication configuration.
- **PLAINTEXT**: Use Kafka plain text authentication. You must provide a **PRODUCER_USERNAME** and **PRODUCER_PASSWORD** with the authentication information for the producer to write to the external Kafka sink.
- **SASL**: Use SASL based authentication between the IBM Spectrum Scale cluster hosting the clustered watch and the external Kafka sink. You must provide a **PRODUCER_USERNAME** and **PRODUCER_PASSWORD** with the authentication information for the producer to write to the external Kafka sink and you must provide the specific mechanism to use:
 - **SCRAM256**
 - **SCRAM512**
- **CERT**: Use certificate-based authentication and encryption of data in flight between the IBM Spectrum Scale cluster hosting the clustered watch and the external Kafka sink. You must provide an extra three (optional fourth) parameters when specifying this type of authentication for the producer to write to the external Kafka sink:
 - **CA_CERT_LOCATION**: Full path (including the actual file) to the location of the ca-cert file. This field is required.
 - **CLIENT_PEM_CERT_LOCATION**: Full path (including the actual file) to the location of the client certificate (.pem format) file. This field is required.

- **CLIENT_KEY_FILE_LOCATION**: Full path to the location of the client key (client.key) file. This field is required.
- **CLIENT_KEY_FILE_PASSWORD**: Key file password. This field is optional.

For more information about specifying these parameters, see the **mmwatch** command in the *IBM Spectrum Scale: Command and Programming Reference*.

Note: In order for the producer to write to the external Kafka queue, the firewall ports must be open between the source IBM Spectrum Scale cluster and the external Kafka queue.

Note: All of these parameters are used in the **--sink-auth-config** flag of the **mmwatch** command. This parameter is optional. When it is used, you must pass a configuration file with specific parameters. For more information, see the following examples. The first example is of a clustered watch folder setup with SCRAM512 to the external Kafka sink.

```
SINK_AUTH_TYPE:SASL
SINK_AUTH_MECHANISM:SCRAM512
PRODUCER_USERNAME:<will be found in external kafka config>
PRODUCER_PASSWORD:<will be found in external kafka config>
```

The second example is of a CERT-based authentication setup between the IBM Spectrum Scale cluster and the external Kafka sink that the clustered watch folder uses.

```
SINK_AUTH_TYPE:CERT
CA_CERT_LOCATION:<path to certs>
CLIENT_PEM_CERT_LOCATION:<path to pem cert>
CLIENT_KEY_FILE_LOCATION:<path to key>
CLIENT_KEY_FILE_PASSWORD:<password from certificate setup>
```

Clustered watch folder events

Descriptions of the events that are supported in clustered watch folder.

The following events are supported:

Table 17. File access events that are supported by clustered watch folder	
File access event	Notes
IN_ACCESS ²	A file was accessed (read or ran).
IN_ATTRIB ²	Metadata was changed (for example, chmod , chown , setxattr etc.).
IN_CLOSE_NOWRITE	A file or folder that was not opened for writing was closed.
IN_CLOSE_WRITE ²	A file that was opened for writing was closed.
IN_CREATE ²	A file or folder was created in a watched folder.
IN_DELETE ²	A file or folder was deleted from a watched folder.
IN_DELETE_SELF ²	A watched file or folder was deleted.
IN_IGNORED ¹	Event that is triggered when a file system is unmounted. Always follows the IN_UNMOUNT event.
IN_ISDIR ¹	A directory is listed.
IN_MODIFY ²	A file was modified (for example, write or truncate).
IN_MOVE_SELF ²	A folder that was being watched was moved.
IN_MOVED_FROM ²	A file within the watched folder was renamed.

Table 17. File access events that are supported by clustered watch folder (continued)

File access event	Notes
IN_MOVED_TO ²	A file was moved or renamed to this folder.
IN_OPEN	A file or folder was opened.
IN_UNMOUNT ¹	A file system that is being watched is unmounted.

Note:

1. These events are always watched.
2. These events are not applicable to a file system mounted as read-only.

JSON attributes in clustered watch folder

Use this information to learn more about the JSON attributes that are associated with the 12 events that can be specified when enabling a clustered watch and the other events that might be received due to other file system activity.

WF_JSON

The version of the record.

wd

The watch descriptor.

cookie

A unique integer that connects related events. It allows the resulting pair of **IN_MOVED_FROM** and **IN_MOVED_TO** events to be connected.

mask

A hex value representing the event.

event

The event type. One of the following events:

- IN_ACCESS
- IN_ATTRIB
- IN_CLOSE_NOWRITE
- IN_CLOSE_WRITE
- IN_CREATE
- IN_DELETE
- IN_DELETE_SELF
- IN_MODIFY
- IN_MOVED_FROM
- IN_MOVED_TO
- IN_MOVE_SELF
- IN_OPEN

path

The path name of the file that is involved in the event.

clusterName

The name of the cluster where the event took place.

nodeName

The name of the node where the event took place.

nfsClientIp

The IP address of the client involved in the event.

fsName

The name of the file system that is involved in the event.

inode

The inode number of the file that is involved in the event.

filesetID

The fileset ID of the file.

linkCount

The link count of the file.

openFlags

The open flags that are specified during the event (**O_RDONLY**, **O_WRONLY**, **O_RDWR**, **O_CREAT**, etc.) as defined in `fcntl.h`.

poolName

The pool name where the file resides.

fileSize

The current size of the file in bytes.

ownerUserId

The owner ID of the file that is involved in the event.

ownerGroupId

The group ID of the file that is involved in the event.

atime

The time in UTC format of the last access of the file that is involved in the event.

ctime

The time in UTC format of the last inode or metadata change of the file that is involved in the event.

mtime

The time in UTC format of the last time that the file was modified.

eventTime

The time in UTC format of the event.

clientUserId

The user ID of the process that is involved in the event.

clientGroupId

The group ID of the process that is involved in the event.

processId

The process ID that is involved in the event.

bytesRead¹

The bytes read from a file.

bytesWritten¹

The bytes written to a file.

minReadOffset¹

The starting position of bytes read from a file.

maxReadOffset¹

The ending position of bytes read from a file.

minWriteOffset¹

The starting position of bytes written to a file.

maxWriteOffset¹

The ending position of bytes written to a file.

permissions

The permissions on the file that is involved in the event.

acls

The access control lists that are involved in the event (only in case of an acl change event).

xattrs

The extended attributes that are involved in the event (only in case of an Xattr change event).

subEvent

None.

You can start a watch with any of the following events:

- IN_OPEN - 0x20
- IN_ACCESS - 0x01
- IN_MODIFY - 0x02
- IN_ATTRIB - 0x04
- IN_CLOSE_WRITE - 0x8
- IN_CLOSE_NOWRITE - 0x10
- IN_CREATE - 0x100
- IN_DELETE - 0x200
- IN_MOVED_FROM - 0x40
- IN_MOVED_TO - 0x80
- IN_DELETE_SELF - 0x400
- IN_MOVE_SELF - 0x800

You might see these events during some of the previously mentioned watch types:

- IN_IGNORED - 0x8000
- IN_UNMOUNT - 0x2000
- IN_OPEN IN_ISDIR - 0x40000020
- IN_ATTRIB IN_ISDIR - 0x40000004
- IN_CLOSE IN_ISDIR - 0x40000010
- IN_CREATE IN_ISDIR - 0x40000100
- IN_DELETE IN_ISDIR - 0x40000200
- IN_MOVE_SELF IN_ISDIR - 0x40000800
- IN_MOVED_FROM - 0x40000040
- IN_MOVED_TO - 0x40000080
- IN_DELETE_SELF - 0x40000400

Note: ¹ Results might be inaccurate with mmap IO, mmap IO via SMB, or IO via NFS.

Understanding call home

IBM Spectrum Scale users can use the call home feature to share the basic configuration data of a cluster with IBM support and development team. The shared details ensure that the cluster is configured to function according to its maximum potential.

With the call home feature enabled, the IBM Spectrum Scale users can upload the daily or weekly cluster configuration or debugging data from the customer site to a specific IBM server. The IBM server then sends the data to the IBM backend, Enhanced Customer Data Repository (IBM ECuRep). For more information, see [ECuRep](#).

The uploaded data is further analyzed by the IBM support team to determine whether there has been a common misconfiguration, best practice violation, or any other issues. This enables the IBM support team to provide solutions to any problems encountered on the customer site or leverage the data to provide additional benefits to the customer. For more information on understanding the call home benefits and

supported use cases, see *Benefits of enabling call home in IBM Spectrum Scale: Problem Determination Guide*.

The call home component in IBM Spectrum Scale must be configured before it can be used for manual or automated data uploads to the IBM server. The call home configuration process consists of the following steps:

1. Configuring the call home settings.
2. Creating call home groups.

For more information, see *Configuring call home* in the *IBM Spectrum Scale: Administration Guide*.

Call home architecture

The following figure shows the basic call home group structure:

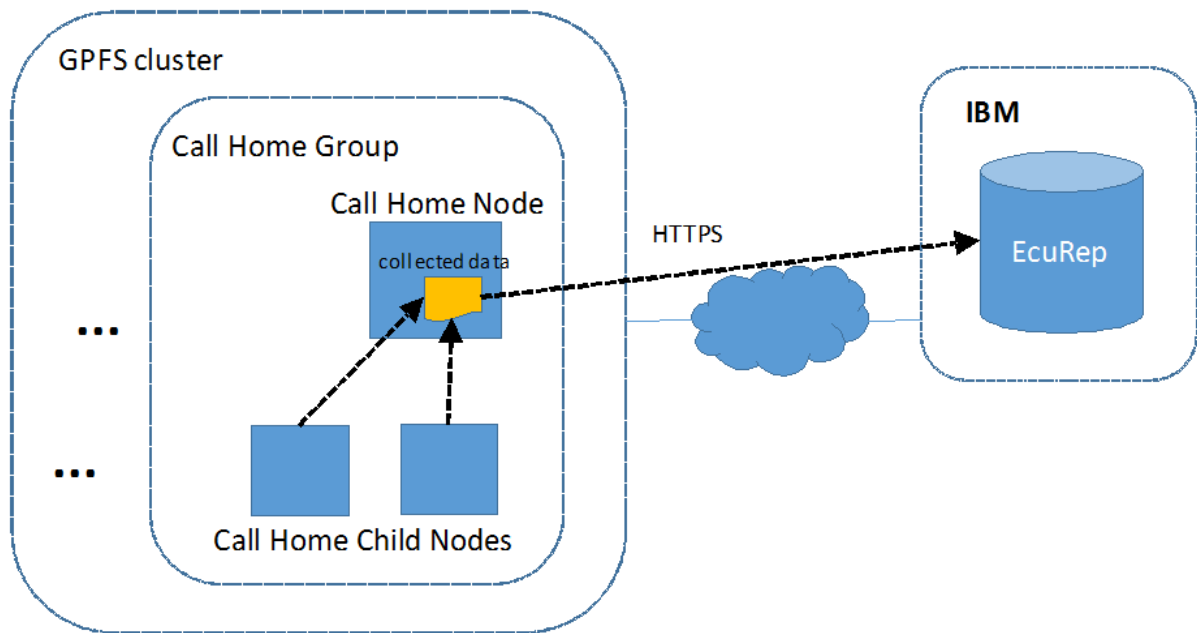


Figure 26. Call home architecture

Call Home Group

The Call Home Group is a group of call home nodes that are configured by using the **mmcallhome group** command. A call home group consists of at least one child node, which also acts as its call home node. A call home group can have more than one child node but has only one call home node. Multiple call home groups can be configured within an IBM Spectrum Scale cluster. You can automate the call home group creation by using the **mmcallhome group auto** command.

Call home groups help to distribute the data-gather and data-upload workload to prevent bottlenecks. You can create groups of any size, and the size might vary between one and the number of nodes in your cluster. The larger the group, the higher the workload on the call home node. Therefore, you are recommended to limit the group size to 1024 nodes. Larger groups are possible, but it might lead to performance issues on the call home node.

Call Home Node

The Call Home Node is the node that performs the data upload. If a scheduled data gathering is enabled, then the call home node initiates the data collection within its call home group and uploads the data package to the IBM support centre. A gather-send task runs on the call home node, which collects the data from the call home child nodes and uploads that data to a specific IBM server. The IBM server then sends the data to the IBM backend, ECuRep. For more information, see [ECuRep](#).

The gather-send configuration file includes information about the data that is collected from the child nodes.

Note: The call home node is also a child node of the group. If the call home node becomes unavailable, then the whole call home group is unable to perform any data uploads until the call home node is online again.

Important: The call home node needs to have access to the external network through 443 port number. For more information on network-related requirements, see the *Installing call home* section in the *IBM Spectrum Scale: Concepts, Planning, and Installation Guide*.

Call Home Child Node

The Call Home Child Node is a member of a call home group. The call home node can collect data from all the call home child nodes in a call home group.

Note: If a call home child node, which is not a call home node, becomes unavailable, then the rest of the call home group continues to work. However, the scheduled uploads do not include the details of the unavailable child node.

mmcallhome commands impact

The call home data upload and share are managed by using the **mmcallhome** command.

The **mmcallhome** command provides options to configure, enable, run, schedule, and monitor call home related tasks in the IBM Spectrum Scale cluster. Information from each node within a call home group is collected and securely uploaded to the IBM ECuRep server.

The uploaded packages that contain the daily or weekly scheduled uploads and non-ticket-related sent files are saved on the corresponding call home node for an inspection by the customer.

You can use the **mmcallhome status list --verbose** command to find the exact location of the uploaded packages. A maximum of 1 GB of space is used by the packages of each of these categories:

- Sent packages
- Failed uploads

A maximum of 48 such packages can be kept.

The uploaded data is stored for at least two weeks on IBM ECuRep, and can be identified by using your customer information. If you need to access this data, contact IBM support. For more information, see [ECuRep](#). The PMR or Salesforce case-related data is saved and the PMR or Salesforce case is still open.

The following **mmcallhome** command options respond in the same way when run on nodes, which belong or do not belong to a call home group:

- **mmcallhome group**
- **mmcallhome capability**
- **mmcallhome info**
- **mmcallhome proxy**

Note: For compatibility reasons, in a mixed cluster configuration, the **mmcallhome capability**, **mmcallhome info**, and **mmcallhome proxy** commands are available only to the global settings if the corresponding nodes are not part of a call home group. If the call home node of this group also has IBM Spectrum Scale 4.2.3 PTF 6 or older nodes, then they also manage a separate settings configuration for their group.

All other mmcallhome command options

All other **mmcallhome** commands can be run only from a node, which is a member of a call home group.

For more information, see the *mmcallhome command* in the *IBM Spectrum Scale: Command and Programming Reference*.

Types of call home data upload

The call home feature allows data to be uploaded either manually or automatically to the IBM server.

The manual and automatic data upload options can upload the data to the IBM ECuRep server. The data which is uploaded to the IBM ECuRep server is not analyzed automatically and gets deleted after a specified period. The default period is set to two weeks, but it can be changed. For more information about the usage of the uploaded data, contact IBM support.

Note: You must consider the space requirements of your system before the call home data is uploaded. For more information, see *Space requirements for call home data upload* in the *IBM Spectrum Scale: Concepts, Planning, and Installation Guide*.

Manual data upload

The call home feature provides the option to manually upload a specific file or multiple data packages to the IBM server. To upload data manually, issue the **mmcallhome run** command in one of the following ways:

- To manually initiate the **daily data** upload, issue the following command:

```
mmcallhome run GatherSend --task DAILY
```

- To manually upload a **specific data file** to the IBM server, issue one of the following commands:

```
- mmcallhome run SendFile --file myfile
```

The **mmcallhome run SendFile --file <file name>** command uploads a specific file to a common directory, which is available for support to all customers.

Note: The IBM support team must be informed that a specific file is to be considered and all IBM support representatives can read the specific file.

```
- mmcallhome run SendFile --file myFile --pmr TS123456789
```

The **mmcallhome run SendFile --file <file name> -pmr <number>** command uploads a specific file to a specific location, which is linked to the specified PMR or Salesforce team.

Note: The IBM support team automatically notices the changes in a PMR or Salesforce case if they are working on the same ticket. Only a specific set of IBM support representatives who are allowed to work on the specific PMR or Salesforce case can read the contents of the specific file.

Automatic data upload

The call home feature provides three different ways to automatically upload a specific file or data packages to the IBM server.

- **Event-based data upload**

This type of data upload collects the debugging information after an occurrence of a specific failure is detected. For more information, see [“Event-based uploads” on page 192](#).

- **Scheduled data upload**

This type of data upload is enabled by using the **mmcallhome schedule** command to schedule a weekly or daily upload of the predefined data. For more information, see [“Scheduled data upload” on page 194](#).

- **Heartbeats feature**

This type of data upload is done only once a day by each call home master node. The call home master nodes send the basic cluster information to IBM ECuRep server. For more information, see [“Heartbeats feature for call home” on page 194](#).

Event-based uploads

If the call home feature is enabled and one of the specific RAS events occur, which degrades the current state of an **mmhealth** component, then the corresponding debugging data is collected and automatically uploaded to IBM ECuRep server for a detailed problem analysis. This event-based data upload feature is called **FTDC2CallHome**.

The **FTDC2CallHome** feature provides the following benefits:

- Allows the IBM support representatives to receive and analyze the relevant debugging data in a faster and easier manner, which reduces the duration of detected outages.
- Directs the efforts of IBM support representatives to the areas that are facing the maximum outages. This increases the stability of IBM Spectrum Scale in the areas that face outage issues more often.
- Introduces features for protection against data flooding. In this way, this feature ensures minimal CPU usage and preserves the network bandwidth, which might be needed to upload the debugging data if any issue occurs.

You can run the following command to see the uploaded data files:

```
mmcallhome status list --task sendfile --verbose
```

Note: If you disable the event-based uploads, then the unified call home feature for Elastic Storage Server (ESS) systems is also disabled to ensure that service tickets are not automatically created for any hardware failures.

The following RAS events trigger data collection:

Table 18. RAS events that trigger data collection	
Event name	Snap
ads_down	auth.snap.py
ads_failed	auth.snap.py
ccr_auth_keys_fail	commonsnap.py
ccr_comm_dir_fail	commonsnap.py
ccr_paxos_12_fail	commonsnap.py
ces_many_tx_errors	commonsnap.py
ces_network_ips_not_assignable	ces.snap.py
ctdb_down	smb.snap.py
ctdb_state_down	smb.snap.py
ctdb_version_mismatch	smb.snap.py
fserrallocblock	commonsnap.py
fserrbadacref	commonsnap.py
fserrbadirblock	commonsnap.py
fserrbaddiskaddrindex	commonsnap.py
fserrbaddiskaddrsector	commonsnap.py
fserrbaddittoaddr	commonsnap.py
fserrbadinodeorgen	commonsnap.py
fserrbadinodestatus	commonsnap.py
fserrbadptrreplications	commonsnap.py

Table 18. RAS events that trigger data collection (continued)

Event name	Snap
fserrbadreplicationcounts	commonsnap.py
fserrbadxattrblock	commonsnap.py
fserrcheckheaderfailed	commonsnap.py
fserrclonetree	commonsnap.py
fserrdeallocblock	commonsnap.py
fserrdotdotnotfound	commonsnap.py
fserrgennummismatch	commonsnap.py
fserrinconsistentfilesetrootdir	commonsnap.py
fserrinconsistentfilesetsnapshot	commonsnap.py
fserrinconsistentinode	commonsnap.py
fserrindirectblock	commonsnap.py
fserrindirectionlevel	commonsnap.py
fserrinodecorrupted	commonsnap.py
fserrinodenummismatch	commonsnap.py
fserrinvalid	commonsnap.py
fserrinvalidfilesetmetadataarecord	commonsnap.py
fserrinvalidsnapshotstates	commonsnap.py
fserrsnapinodemodified	commonsnap.py
fserrvalidate	commonsnap.py
kafka_failed	msgqueue.snap.py
ks_failed	object.snap.py
many_tx_errors	commonsnap.py
mmfsd_abort_warn	commonsnap.py
nfsd_down	nfs.snap.py
nfsd_restart	nfs.snap.py
postgresql_failed	object.snap.py
proxy-server_failed	object.snap.py
ring_checksum_failed	object.snap.py
smbd_down	smb.snap.py
zookeeper_failed	msgqueue.snap.py

Heartbeats feature for call home

If the call home feature is installed and enabled on the IBM Spectrum Scale cluster, then the CALLHOME service runs on the call home master nodes. As soon as the CALLHOME service is enabled, it starts collecting heartbeat data once a day.

The heartbeats feature uploads a snap of the basic cluster information to the IBM ECuRep server. The call home schedule settings do not affect the heartbeats feature.

The basic snap script, `basicsnap.py` collects only the following basic cluster information:

- `mmdia -v -Y`
- `mmsysmon d cfigshow`
- `df -k -t gpfs`
- `lsblk`
- `lscpu`
- `/etc/os-release`
- `/proc/cpuinfo`
- `/proc/meminfo`
- `/proc/uptime`
- `mmlslicense -Y`
- `mmlsconfig -Y`

Note: The CALLHOME component is a part of the `mmhealth` command monitoring list. For more information, see the `mmhealth` command in the *IBM Spectrum Scale: Command and Programming Reference* guide, and the *Call home events* section, and *Monitoring the health of a node* section in the *IBM Spectrum Scale: Problem Determination Guide*.

Scheduled data upload

You can run the `mmcallhome schedule` command to scheduled data uploads.

The call home feature has a built-in data collection mechanism that collects predefined data on a daily or weekly basis. You can find the schedules that are defined for this data collection in `/usr/lpp/mmfs/lib/mmsysmon/callhome/callhomeSchedules.json`.

The following data is collected and uploaded regularly by the call home feature on a daily or weekly basis according to the defined schedule:

Table 19. Data collected and uploaded by call home					
COMMAND	MACHINE-TYPE	NODE	OS	PRODUCT	SCHEDULE
<code>cat /etc/ganesha/config/gpfs.ganesha.main.conf</code>	all	all	all	all	all
<code>cat /etc/os-release</code>	all	all	all	all	all
<code>cat /opt/ibm/esa/data/IBM_ESAconfig.properties</code>	all	CALLHOME_SERVERS	RHEL, SLES	ess	all
<code>cat /proc/cpuinfo</code>	all	all	all	all	all
<code>cat /proc/device-tree/system-id</code>	power	all	all	all	all

Table 19. Data collected and uploaded by call home (continued)

COMMAND	MACHINE-TYPE	NODE	OS	PRODUCT	SCHEDULE
cat /proc/interrupts	all	all	all	all	all
cat /proc/meminfo	all	all	all	all	all
cat /proc/net/dev	all	all	all	all	all
cat /var/mmfs/gen/mmfs.cfg.show	all	CALLHOME_SERVERS	all	all	all
/containermon/callhome/gui.callhome.py <lcdir>	all	all	all	all	all
/containermon/callhome/perfmon.container.callhome.py <lcdir>	all	all	all	all	all
/containermon/get_ganesha_stats.py	all	all	all	all	all
df -kl	all	all	all	all	all
dmidecode	x86	all	all	all	all
dpkg-query --list gpfs.compression gpfs.gui gpfs.adv gpfs.msg.en-us gpfs.java gpfs.license.adv gpfs.crypto gpfs.docs gpfs.gskit gpfs.gpl gpfs.base	all	all	DEBIAN	all	all
dpkg-query --list gpfs.nfs-ganesha gpfs.nfs-ganesha-doc gpfs.nfs-ganesha-gpfs nfs-ganesha nfs-ganesha-doc nfs-ganesha-fsal python-nfs-ganesha gpfs.smb gpfs.gss.pmsensors pmswift gpfs.pm-ganesha gpfs.gss.pmc collector	all	all	DEBIAN	all	all

Table 19. Data collected and uploaded by call home (continued)

COMMAND	MACHINE-TYPE	NODE	OS	PRODUCT	SCHEDULE
<code>dpkg-query --list openstack-utils openstack- python-swift openstack-swift openstack-swift-object openstack-swift-account python-keystoneclient python- keystonemiddleware openstack- swift-proxy spectrum-scale- object python-swiftclient python-keystoneauth1 python-openstackclient openstack-swift-container swiftonfile openstack-swift- object-expirer openstack- python-keystone openstack- keystone swift3</code>	all	all	DEBIAN	all	weekly
<code>echo -e "ver\nio_s\ nfs_io_s\nvio_s\ nrhist_s" /usr/lpp/mmfs/ bin/mmpmon 2> /dev/null</code>	all	CALLHOME _SERVERS	all	all	all
<code>echo /install/?ss/manifest*; cat /install/?ss/manifest*</code>	all	all	all	ess	all
<code><GET_FILES_READABLE_BY_ALL> /var/mmfs/ccr/committed/*</code>	all	all	all	all	all
<code><GET_FILES_READABLE_BY_ALL> / var/mmfs/hadoop/etc/hadoop/*- env.sh</code>	all	cesNodes	all	all	all
<code><GET_FILES_READABLE_BY_ALL> / var/mmfs/hadoop/etc/hadoop/*- site.xml</code>	all	cesNodes	all	all	all
<code>ibdev2netdev</code>	all	all	all	all	all
<code>ibstat</code>	all	all	all	all	all
<code>ip a</code>	all	all	all	all	all
<code>ip route</code>	all	all	all	all	all
<code>journalctl -k -p 5 -n 10000 -r --since="yesterday"</code>	all	all	all	all	all

Table 19. Data collected and uploaded by call home (continued)

COMMAND	MACHINE-TYPE	NODE	OS	PRODUCT	SCHEDULE
<code>ls -l /sys/class/sas_host sort -k1.5 -n while read a ; do echo "# \$a" ; cat /sys/ class/sas_host/\${a}/device/ scsi_host/\${a}/version* ; done</code>	all	all	all	all	all
<code>lsblk</code>	all	all	all	all	weekly
<code>lscpu</code>	all	all	all	all	all
<code>lscss --vpm --avail</code>	s390x	all	all	all	all
<code>lsdasd -a -u</code>	s390x	all	all	all	all
<code>lsdasd -b -l</code>	s390x	all	all	all	all
<code>lsmod</code>	all	all	all	all	all
<code>lspci -vv</code>	all	all	all	all	weekly
<code>lsqeth -p</code>	s390x	all	all	all	all
<code>lsscsi -li</code>	all	all	all	all	all
<code>lszfcp -H -P -D</code>	s390x	all	all	all	all
<code>multipath -ll</code>	all	all	all	all	all
<code>nstat -asj</code>	all	all	all	all	all
<code>numactl --hardware</code>	all	all	all	all	weekly
<code>numastat -nm</code>	all	all	all	all	weekly
<code>ofed_info -s</code>	all	all	all	all	all
<code>/opt/ibm/esa/bin/isactivated</code>	all	CALLHOME _SERVERS	RHEL , SLES	ess	all
<code>/opt/ibm/ess/tools/bin/ ess3kplt --local -Y</code>	all	all	all	ess3000	weekly

Table 19. Data collected and uploaded by call home (continued)

COMMAND	MACHINE-TYPE	NODE	OS	PRODUCT	SCHEDULE
ppc64_cpu --smt;ppc64_cpu --cores-present;ppc64_cpu --cores-on	power	all	all	all	all
route -n	all	all	all	all	all
rpm -qi esagent.pLinux	all	CALLHOME_SERVERS	RHEL, SLES	ess	all
rpm -qi gpfs.ess.tools gpfs.ess.firmware	all	all	RHEL, SLES	ess	all
rpm -qi gpfs.license.adv gpfs.gui gpfs.kafka gpfs.scst gpfs.librdkafka gpfs.gpl gpfs.tct.client gpfs.msg.en_US gpfs.base gpfs.java gpfs.docs gpfs.adv gpfs.compression gpfs.gskit gpfs.hdfs- protocol gpfs.tct.server gpfs.crypto	all	all	RHEL, SLES	all	all
rpm -qi gpfs.smb gpfs.gss.pmcollector gpfs.gss.pmsensors pmswift gpfs.pm-ganesha gpfs.nfs- ganesha gpfs.nfs-ganesha-doc gpfs.nfs-ganesha-gpfs nfs- ganesha nfs-ganesha-gpfs nfs- ganesha-utils	all	all	RHEL, SLES	all	all

Table 19. Data collected and uploaded by call home (continued)

COMMAND	MACHINE-TYPE	NODE	OS	PRODUCT	SCHEDULE
<pre>rpm -qi python-cinderclient python-testtools python- cryptography openstack-swift- account python-ipaddress spectrum-scale-object python- iso8601 python-keystone python-fixtures python-kombu python-repoze-who python- rfc3987 python-fasteners python-oslo-service python- zope-interface python- pyasn1 python-wsgiref python-vcversioner python- dnspython python-cadf python-requestsexceptions python-cliff-tablib python- netifaces python-idna python- posix_ipc python-pysaml2 python-prettytable openstack- swift swift3 python- openstackclient python- greenlet python-pyeclib pyparsing python-requests python-futures python- pycparser python-netaddr python-stevedore python- swiftclient python-alembic python-paste-deploy libcap- ng-python python-msgpack python-oslo-config python- argparse python-anyjson keystonemiddleware python- jinja2</pre>	all	all	RHEL , SLES	all	weekly
<pre>rpm -qi python-monotonic python-positional python- aioeventlet python-oslo- messaging openstack-swift- object python-jsonschema python-novaclient python- oslo-context python-dogpile- cache isa-l python-os- client-config python-babel python-oslo-cache python- openstacksdk python-pika PyYAML liberasurecode python- keyring python-markupsafe python-ldappool openstack- utils python-editor python- extras python-strict-rfc3339 python-oauthlib python- funcsigs python-oslo-i18n python-trollius openstack- swift-proxy python-tablib python-sqlparse python- pika_pool python-eventlet python-keystoneauth1 python- contextlib2 numpy python- pbr python-cachetools python- httplib2 openstack-selinux python-oslo-log python- ordereddict openstack-swift- container python-webcolors</pre>	all	all	RHEL , SLES	all	weekly

Table 19. Data collected and uploaded by call home (continued)

COMMAND	MACHINE-TYPE	NODE	OS	PRODUCT	SCHEDULE
rpm -qi python-oslo-db python-retrying pyxattr python-amqp python-mimeparse python-cmd2 swiftonfile pytz python-pip spectrum-scale- object-selinux python-cffi babel python-urllib3 python- rfc3986 python-functools32 python-wrapit python-ply python-warlock python- passlib python-webob python- appdirs python-dogpile-core python-jsonpointer python- keystoneclient crudini python-debtcollector python- oslo-utils python-crypto python-simplejson libcap- ng xmlsec1 python-oslo- concurrency python-futurist openstack-keystone python- six python-cliff python-zope- event python-oslo-middleware python-lxml python-PyMySQL python-oslo-serialization python-jsonpatch python- enum34 python-dateutil python-unicodcsv python- sqlalchemy python-routes python-glanceclient python- oslo-policy python-migrate python-inotify python- neutronclient	all	all	RHEL, SLES	all	weekly
sysctl -a	all	all	all	all	all
tail -n 10000 /var/adm/ras/mmaudit.log	all	all	all	all	all
tail -n 10000 /var/adm/ras/ mmfs.log.latest	all	all	all	all	all
tail -n 1000 /var/adm/ras/mmsysmonitor.log	all	all	all	all	all
tail -n 10000 /var/adm/ras/mmwf.log	all	all	all	all	all
uname -a	all	all	all	all	all
uptime	all	all	all	all	all
/usr/lpp/mmfs/bin/ ece.callhome.py	all	all	all	ess	all
/usr/lpp/mmfs/bin/ gui.callhome.py <lcdir>	all	all	all	all	all

Table 19. Data collected and uploaded by call home (continued)

COMMAND	MACHINE-TYPE	NODE	OS	PRODUCT	SCHEDULE
/usr/lpp/mmfs/bin/mmaudit all functional --list-all- filesystems-config	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/ mmauth show -Y	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/ mmccr vlist	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/ mmces address list -Y	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/mmces service list --verbose -Y	all	CALLHOME _SERVERS	all	all	weekly
/usr/lpp/mmfs/bin/mmdf <fs> -Y	all	CALLHOME _SERVERS	all	all	weekly
/usr/lpp/mmfs/bin/mmdiag --afm -Y	all	all	all	all	all
/usr/lpp/mmfs/bin/mmdiag --commands -Y	all	all	all	all	all
/usr/lpp/mmfs/bin/ mmdiag --config -Y	all	all	all	all	weekly
/usr/lpp/mmfs/bin/mmdiag --deadlock -Y	all	all	all	all	all
/usr/lpp/mmfs/bin/mmdiag --eventproducer -Y	all	all	all	all	all
/usr/lpp/mmfs/bin/mmdiag --health -Y	all	all	all	all	all
/usr/lpp/mmfs/bin/mmdiag --lroc -Y	all	all	all	all	all
/usr/lpp/mmfs/bin/mmdiag --memory -Y	all	all	all	all	all
/usr/lpp/mmfs/bin/mmdiag --network -Y	all	all	all	all	all
/usr/lpp/mmfs/bin/mmdiag --nsd -Y	all	all	all	all	all

Table 19. Data collected and uploaded by call home (continued)

COMMAND	MACHINE-TYPE	NODE	OS	PRODUCT	SCHEDULE
/usr/lpp/mmfs/bin/mmdiag --rpc 24h -Y	all	all	all	all	all
/usr/lpp/mmfs/bin/mmdiag --stats -Y	all	all	all	all	all
/usr/lpp/mmfs/bin/mmdiag --tokenmgr -Y	all	all	all	all	all
/usr/lpp/mmfs/bin/ mmdiag --version -Y	all	all	all	all	all
/usr/lpp/mmfs/bin/mmdiag --waiters -Y	all	all	all	all	all
/usr/lpp/mmfs/bin/mmfsadm dump pdisk	all	all	all	ess	weekly
/usr/lpp/mmfs/bin/ mmhealth cluster show --verbose -Y	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/ mmhealth config interval -Y	all	all	all	all	all
/usr/lpp/mmfs/bin/ mmhealth event list hidden -Y	all	all	all	all	all
/usr/lpp/mmfs/bin/ mmhealth node eventlog --day -Y	all	all	all	all	daily
/usr/lpp/mmfs/bin/ mmhealth node eventlog --week -Y	all	all	all	all	weekly
/usr/lpp/mmfs/bin/mmhealth node show -v -Y	all	all	all	all	all
/usr/lpp/mmfs/bin/ mmhealth thresholds list -Y	all	all	all	all	all
/usr/lpp/mmfs/bin/ mmlscallback -Y	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/ mmlscluster --ces -Y	all	CALLHOME _SERVERS	all	all	all

Table 19. Data collected and uploaded by call home (continued)

COMMAND	MACHINE- TYPE	NODE	OS	PRODUCT	SCHEDULE
/usr/lpp/mmfs/bin/ mmlscluster --cnfs -Y	all	CALLHOME _SERVERS	all	all	weekly
/usr/lpp/mmfs/bin/ mmlscluster -Y	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/ mmlscomp -Y	all	CALLHOME _SERVERS	all	ess	all
/usr/lpp/mmfs/bin/ mmlsenclosure all -Y -N GUI_RG_SERVERS	all	CALLHOME _SERVERS	all	ess	all
/usr/lpp/mmfs/bin/ mmlsfileset <fs> -L -Y	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/ mmlsfirmware -Y	all	CALLHOME _SERVERS	all	ess	weekly
/usr/lpp/mmfs/bin/ mmlsfs <fs> -aABdDEfiIkKLnoPQStTuVwYz --create-time --encryption --fastea --filesetdf --inode-limit --is4KAligned --log-replicas --mount-priority --perfilesset --rapid-repair --write-cache-threshold --striped-logs --fileset-count --afm --snc --uid --snapid	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/ mmlslicense -L -Y	all	CALLHOME _SERVERS	all	all	weekly
/usr/lpp/mmfs/bin/ mmlsmgr -Y	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/ mmlsnode -a	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/ mmlsnodeclass --all -Y	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/ mmlsnsd -L -Y	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/ mmlspdisk all -Y	all	CALLHOME _SERVERS	all	ess	all

Table 19. Data collected and uploaded by call home (continued)

COMMAND	MACHINE-TYPE	NODE	OS	PRODUCT	SCHEDULE
/usr/lpp/mmfs/bin/ mmlspool <fs> all -Y	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/ mmlsrecoverygroupevents <rg> --days 2	all	CALLHOME _SERVERS	all	ess	daily
/usr/lpp/mmfs/bin/ mmlsrecoverygroupevents <rg> --days 8	all	CALLHOME _SERVERS	all	ess	weekly
/usr/lpp/mmfs/bin/ mmlsrecoverygroup -Y	all	CALLHOME _SERVERS	all	ess	all
/usr/lpp/mmfs/bin/ mmlssnapshot <fs> -Y	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/ mmlsvdisk -Y	all	CALLHOME _SERVERS	all	ess	all
/usr/lpp/mmfs/bin/mmfs config list -Y	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/mmfs export list -Y	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/mmqs device status <fs> -Y	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/mmqs report list <fs> -Y	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/ mmremoteccluster show all -Y	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/ mmremotefs show all -Y	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/mmsmb config list -Y	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/mmsmb export list --all -Y	all	CALLHOME _SERVERS	all	all	all
/usr/lpp/mmfs/bin/ mmsdrquery sdrq_fs_info all	all	CALLHOME _SERVERS	all	all	all

Table 19. Data collected and uploaded by call home (continued)

COMMAND	MACHINE-TYPE	NODE	OS	PRODUCT	SCHEDULE
/usr/lpp/mmfs/bin/mmsysmonc d cfgshow	all	all	all	all	all
/usr/lpp/mmfs/bin/mmuserauth service list -Y	all	CALLHOME_SERVERS	all	all	all
/usr/lpp/mmfs/bin/mmwatch all functional --list-clustered-watch-config	all	CALLHOME_SERVERS	all	all	all
/usr/lpp/mmfs/bin/network.callhome.py <lcdir>	all	all	all	all	all
/usr/lpp/mmfs/bin/object.callhome.py	all	CALLHOME_SERVERS	all	all	all
/usr/lpp/mmfs/bin/perfmon.callhome.py <lcdir>	all	all	all	all	daily
/usr/lpp/mmfs/bin/tsctl nqStatus -Y	all	all	all	all	all
/usr/lpp/mmfs/bin/tlsencslot -adY	all	CALLHOME_SERVERS	all	ess	all
/usr/lpp/mmfs/bin/tlspolicy <fs> -L -Y --ptn	all	CALLHOME_SERVERS	all	all	all
/usr/lpp/mmfs/bin/tsstatus	all	CALLHOME_SERVERS	all	all	all
/usr/lpp/mmfs/lib/mmsysmon/container/openshift.collect.py Pod DaemonSet Deployment StatefulSet	all	CALLHOME_SERVERS	all	all	all
/usr/lpp/mmfs/lib/mmsysmon/noobaa_api.py noobaa_call_home > <lcdir>/noobaa_data.txt	all	all	all	all	all
/usr/lpp/mmfs/samples/vdisk/gnrhealthcheck	all	CALLHOME_SERVERS	all	ess	weekly
/usr/lpp/mmfs/bin/tsplatformstat -a	all	all	all	ess3000	all

Inspecting call home data uploads

The uploaded call home data is generally used by the IBM service and support teams for diagnosing issues and best practice violations, which can be reported at a customer site. However, a customer can opt to self-inspect a sample call home data upload to manually review the data without sending out any information to IBM.

You can inspect call home data uploads without sending out any information to the IBM by configuring the call home feature by using the following procedure:

1. Set up the call home feature but provide incorrect proxy settings.

For example,

```
[root@g5130-11 ~]# mmcallhome proxy change --proxy-location wrong_proxy --proxy-port 1111
Call home proxy-location has been set to wrong_proxy
Call home proxy-port has been set to 1111

[root@g5130-11 ~]# mmcallhome proxy enable
Call home proxy-auth-enabled has been set to false
Call home proxy-enabled has been set to true
```

2. Start the uploaded data collection, such as daily uploaded data. The daily data upload overrides the standard 24-hour wait time and delivers the results immediately.

For example,

```
[root@g5130-11 ~]# mmcallhome run GatherSend --task daily
One time run completed with failure: sending failed
```

As expected, the data uploads to the server fails as you provided incorrect proxy settings.

3. Find the DC file with the copy of the data that the call home feature collected and tried to send.

For example,

```
[root@g5130-11 ~]# mmcallhome status list --task daily --verbose --numbers 1
=== Executed call home tasks ===

Group      Task      Start Time          Updated Time        Status  RC or Step
Package File
Name
          Original Filename
-----
autoGroup_1 daily  20220126153252.022  20220126153310    failed  Uploading operation
failed /tmp/mmfs/callhome/rsENFailedQ/
14559237835643.5_1_3_0.123456.US.ibmtest.autoGroup_1.gat_daily.g_daily.scale.2022012615325202
2.c10.DC
```

4. Extract the DC file, which is a *.tar.gz file.

For example,

```
[root@g5130-11 ~]# mkdir wrong_proxy
[root@g5130-11 ~]# cd wrong_proxy/
[root@g5130-11 wrong_proxy]# tar xvf /tmp/mmfs/callhome/rsENFailedQ/
14559237835643.5_1_3_0.123456.US.ibmtest.autoGroup_1.gat_daily.g_daily.scale.2022012615325202
2.c10.DC
CH_20220126153252.022_daily/
CH_20220126153252.022_daily/HEADER
CH_20220126153252.022_daily/LC_g5130-14.tar.gz
CH_20220126153252.022_daily/LC_g5130-13.tar.gz
CH_20220126153252.022_daily/LC_g5130-12.tar.gz
CH_20220126153252.022_daily/MC_g5130-11.tar.gz
[root@g5130-11 wrong_proxy]# tar xvf CH_20220126153252.022_daily/MC_g5130-11.tar.gz
MC_g5130-11/
MC_g5130-11/cat_etcosrelease.txt
...
```

5. Inspect every detail of the DC file.

6. Run the **mmcallhome proxy** command to enable the call home feature and do one of the following actions:

- If you are using a proxy, then provide correct proxy settings for call home usage by using the following command:

```
mmcallhome proxy change --proxy-location <CORRECT_PROXY> --proxy-port <CORRECT_PORT>
```

- If you are using a proxy with authentication, then issue the following command:

```
[root@g5130-11 ~]# mmcallhome proxy enable --with-proxy-auth
Call home proxy-auth-enabled has been set to true
Call home proxy-enabled has been set to true
```

- If you are not using a proxy, then disable the proxy usage for call home by using the following command:

```
[root@g5130-11 ~]# mmcallhome proxy disable
Call home proxy-auth-enabled has been set to false
Call home proxy-enabled has been set to false
```

Benefits of enabling call home

You must configure and enable the call home feature as it improves the user experience when IBM Spectrum Scale is used.

The call home feature when enabled provides the following benefits:

1. Improves customer service response time

- *Scheduled data uploads:*

If the call home feature is enabled, then the IBM support representatives can start with the analysis of the issue that is reported without any delay. The IBM support team can use the cluster data from the call home scheduled uploads to do an immediate analysis of the issue. For more information, see *Scheduled data upload* in the *IBM Spectrum Scale: Concepts, Planning, and Installation Guide*.

However, if the call home feature is not enabled, then the IBM support team first collects debugging data from command outputs like **gpfs.snap** command. The whole process takes time to request, collect, and deliver the data, which might cause a delay in resolving the issue on the site.

- *Selected failure events trigger automatic call home uploads:*

If the call home feature is enabled, then the event-based uploads feature automatically collects the relevant component-specific information and uploads it to the IBM ECuRep server. The uploaded data enables the IBM support team to immediately find the root cause of the issue that is reported on the customer site. For more information, see *Event-based uploads* in the *IBM Spectrum Scale: Concepts, Planning, and Installation Guide*.

2. Proactively detects issues

- Detects violations against best practices and common misconfiguration.
- Finds out which specific customer site is affected by the reported issue by using tools like High Impact Programming Error Authorized Program Analysis Report (HIPER APARs).
- Constantly extends the list of automatically detected issues, best practice violations, and common misconfiguration over time.

Note: The **mmhealth** command output and IBM Spectrum Scale GUI display the list of events that are resulted from proactively detected issues. For more information, see *Proactive system health alerts* in the *IBM Spectrum Scale: Problem Determination Guide*.

3. Improves customer support experience in areas of consumability and ease of usage

- *Uploads gpfs snaps directly from the IBM® Spectrum Scale system:*

If the call home feature is enabled, then a customer can easily find out the customer number. The requested debugging data is transferred from the customer site cluster or node to the IBM support team by using the **mmcallhome** command.

- *Views system configuration changes that take places between call home data uploads:*

If the call home feature is enabled, then you can run the **mmcallhome status diff** command to easily detect the system configuration changes that took place between different time spans.

4. **Generates reports based on customer insights for support and development**

- Improves test and development for the features that are used by your cluster.
- Provides more data to analyze the issues that are reported on the customer site. For example, the root cause of an issue can be diagnosed by using the historic call home data.

5. **Creates service tickets automatically for reported failures**

In Elastic Storage Server (ESS), if the call home feature is enabled, then the system automatically creates Salesforce service tickets for any hardware failures that are reported on customer sites.

Note: A user can configure the call home settings on IBM Spectrum Scale or on Elastic Storage Server (ESS). This feature enhancement is applicable only to Elastic Storage Server (ESS) and automatically opens a Salesforce service ticket to report a hardware issue.

Data privacy with call home

A licensee can configure the IBM Spectrum Scale to automatically send specific cluster information to the IBM support team by using the Call Home feature.

When a licensee enables the Call Home feature, the licensee provides the support contact information, such as names, phone numbers, or email addresses, which are needed during the Call Home feature configuration and activation process.

By default, when a licensee decides to activate and use the Call Home feature, the licensee agrees to allow IBM and its subsidiaries to store and use the licensee's support contact information. This information is processed and used with IBM business relationship and might be shared with third-parties under the direction of IBM. For example, IBM support center representatives or assignees of IBM and its subsidiaries.

The support contact information can be used for processing business orders, business promotions, problem determination, or market research.

When the call home feature is configured and activated, the IBM Spectrum Scale collects all monitoring information that is related to system utilization, performance, capacity planning, and service maintenance. The service information includes system failure logs, part numbers, machine serial number, software version, maintenance levels, installed patches, and configuration values.

When the Call Home feature is enabled, the licensee allows IBM to use and share the data, which is gathered from system monitoring functions, within IBM and with IBM business partners and third-parties, such as IBM support centre sub-contractors or assignees. The shared data is used only under the direction of IBM for the following defined purposes:

- Determining a problem.
- Assisting licensee with performance and capacity planning.
- Assisting IBM to enhance IBM products and services.
- Notifying licensee about the licensee's system status and available solutions.

Note: The licensee information excludes the collection and transmission of licensee's financial, statistical, and personal data, and licensee's business plans.

When the Call Home feature is enabled, a licensee agrees to the fact that the licensee's support contact information can be transferred to countries that might not be a member of the European Union. A licensee can disable the Call Home feature at any time.

Call home monitors for PTF updates

IBM Spectrum Scale provides temporary fixes to its code in between planned releases. The temporary fixes to the codes are called Program Temporary Fix (PTF).

Types of PTF monitors

The call home function has the following two monitors that provide PTF information to a user:

Callhome Update Request

The **Callhome Update Request** monitor is scheduled to run only once a day and exclusively on the first call home master node of a cluster. The monitor requests the PTF update information from the eSupport server.

The monitor output list contains information about the PTF versions that are available for the IBM Spectrum Scale version that is installed on the master node. The PTF information is stored in the Cluster Configuration Repository (CCR) to make it available to all other cluster nodes.

The following events show up during the run time of the **Callhome Update Request** monitor:

callhome_ptfupdates_ok

Informs the user that the PTF update check is processed without any errors.

callhome_ptfupdates_disabled

Informs the user that the call home function is disabled, or the monitor is disabled through configuration.

callhome_ptfupdates_ccr_failed

Informs the user that an error occurred, and the update information cannot be written to the CCR.

callhome_ptfupdates_failed

Informs the user about a connectivity error when a PTF update information is requested from the eSupport server.

callhome_ptfupdates_noop

Informs the user that an event is raised on a call home master node that is not the first node in the list.

Note: When the first call home master node disappears, the next call home master node in the list automatically takes over the master role.

PTF Updates Check

The **PTF Updates Check** monitor is scheduled to run only once a day on all call home child nodes, including the master nodes of the cluster. The monitor reads the update information from the CCR and compares it against the IBM Spectrum Scale version that is installed on the local node. When an appropriate PTF update is found in the information during this check, a system health tip is raised to inform the system administrator about the software update that can be installed.

Note: The software update packages can be downloaded from the Fix Central. However, the software update package download is not done automatically.

The following events show up during the runtime of the **PTF Updates Check** monitor:

scale_up_to_date

Informs the user that the installed packages are up to date. No new PTF packages are available.

scale_updatecheck_disabled

Informs the user that either the call home function or the PTF monitor is disabled by setting the value of **scale_updatecheck_disabled** as true.

scale_ptf_update_available

Informs the user that the PTF updates are available for the current node.

Important:

The call home PTF update notification function can be disabled by changing the configuration value of the **mmhealthPtFUpdatesMonitorEnabled** to no. The **mmhealthPtFUpdatesMonitorEnabled** value can be set by using the following **mmchconfig** command.

```
mmchconfig mmhealthPtFUpdatesMonitorEnabled=no
```

Limitations of PTF monitors

The PTF monitors display the following limitations:

- The two PTF monitors consider only the PTF updates for the major version that is installed on the first call home master node. Other versions that might be installed on the other cluster nodes are not considered.
- The PTF monitors consider only the system architecture of the first call home master node.
- The PTF software update process does not run automatically. You must download the applicable PTF updates from the Fix Central before running them.

IBM Spectrum Scale in an OpenStack cloud deployment

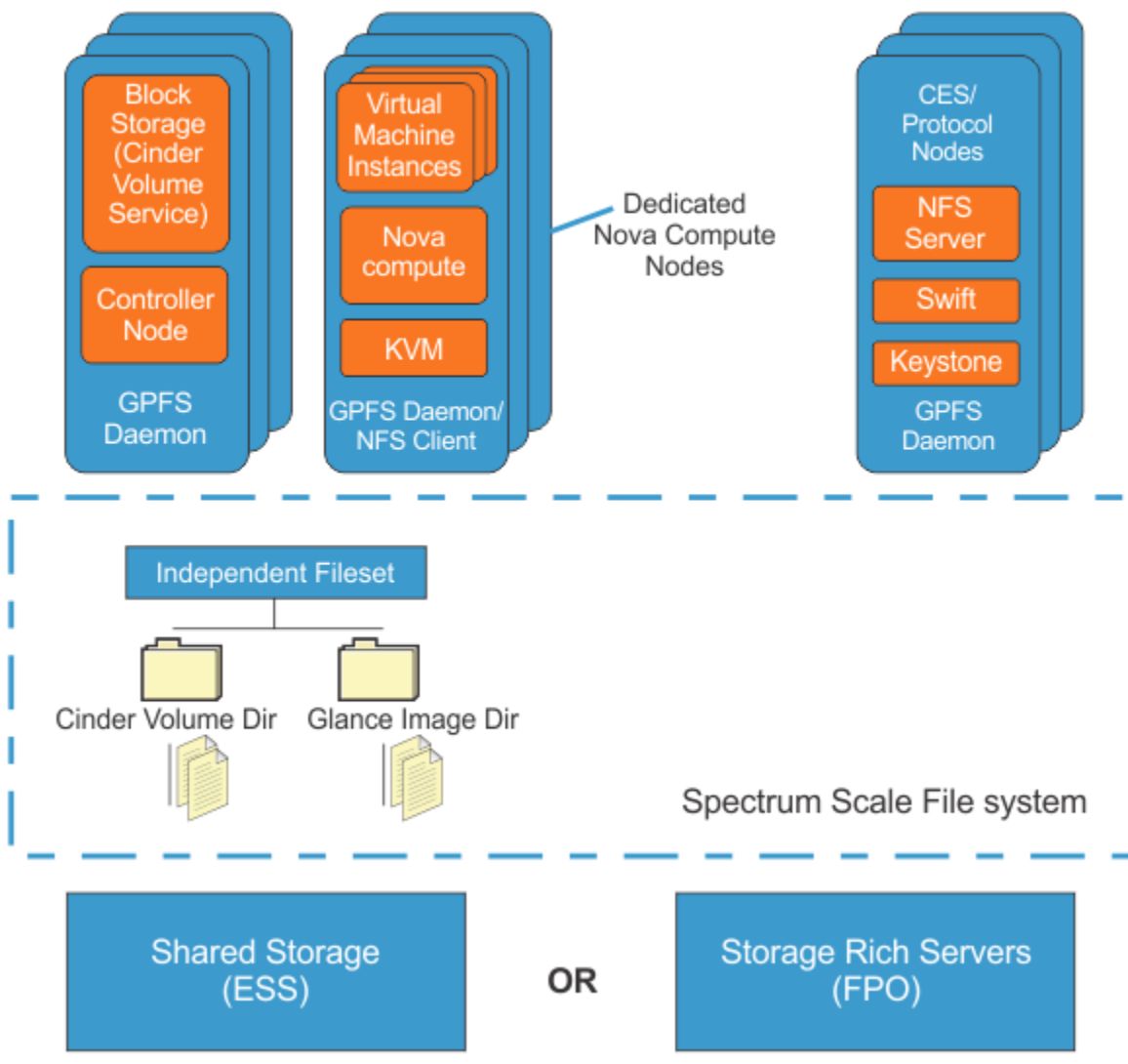
OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter. Using IBM Spectrum Scale in an OpenStack cloud environment offers many benefits, including the enterprise features of IBM Spectrum Scale and consolidated storage options.

An introduction to OpenStack

OpenStack is an open source software platform that is widely used as the base to build cloud infrastructure as a service solution. OpenStack is typically deployed on commodity hardware and is used to virtualize various parts of the infrastructure (compute, storage, and network) to ease the sharing of the infrastructure across applications, use cases and workloads. IBM Spectrum Scale also supports the OpenStack Cinder component with Remote Spectrum Scale Access Deployment mode on Red Hat® OpenStack Platform 16.1.3 onwards. For more information see the topic *Integrating IBM Spectrum Scale Cinder driver with Red Hat OpenStack Platform 16.1* in the *IBM Spectrum Scale: Administration Guide*.

Using IBM Spectrum Scale in an OpenStack cloud deployment

Deploying OpenStack over IBM Spectrum Scale offers benefits provided by the many enterprise features in IBM Spectrum Scale as well as the ability to consolidate storage for various OpenStack components and applications running on top of the OpenStack infrastructure under a single storage management plane. One key benefit of IBM Spectrum Scale is that it provides uniform access to data under a single namespace with integrated analytics.



bl1ins089

OpenStack components supported with IBM Spectrum Scale

- Cinder: Provides virtualized block storage for virtual machines. The IBM Spectrum Scale Cinder driver, also known as the GPFS driver, is written to take full advantage of the IBM Spectrum Scale enterprise features.
- Glance: Provides the capability to manage virtual machine images. When Glance is configured to use the same IBM Spectrum Scale fileset that stores Cinder volumes, bootable images can be created almost instantly by using the copy-on-write file clone capability.
- Swift: Provides object storage to any user or application that requires access to data through a RESTful API. The Swift object storage configuration is optimized for the IBM Spectrum Scale environment, providing high availability and simplified management. Swift object storage also supports native the Swift APIs and Amazon S3 APIs for accessing data. Finally, the Swift object storage also supports access to the same data through either object interface or file interface (POSIX, NFS, SMB) without creating a copy.
- Keystone: Although not a storage component, internal keystone with in-built HA is provided by IBM Spectrum Scale as part of the Object protocol. In deployments that already have keystone support, the Object protocol can be configured to use the external keystone server rather than the internal one.

IBM Spectrum Scale supports OpenStack Cinder component from Ussuri release in OpenStack Community.

The following table lists the available features that IBM Spectrum Scale supports for deploying the OpenStack cloud storage:

<i>Table 20. Features that IBM Spectrum Scale supports for deploying the OpenStack cloud storage</i>	
Feature	Support
Volume and Image Management (Cinder and Glance)	
Volume Creation and Deletion	Yes
Volume Creation from Snapshot	Yes
Image Creation from Volume	Yes
Volume Creation from another Volume	Yes
Live migration of instances	Yes
Encryption of volumes	Yes
Consistency Group Snapshot Creation and Deletion	Yes
Volume Snapshot Creation and Deletion	Yes
Extend (non-attached) volumes	Yes
Consistency Group Creation and Deletion	Yes
Volume Creation from Image	Yes
Instantaneous Boot Volume Create From Glance Repo	Yes
Backup of volumes	Yes
Quality of service using multi-tier storage (with Flash support)	Yes
Tiering of volumes	Yes
Volume Attach and Detach to a VM instance	Yes
Instantaneous Boot Volume Create From Glance Repo	Yes
Identity Management (Keystone)	
Integrated High Availability across the IBM Spectrum Scale protocol nodes	Yes
AD and LDAP Support	Yes
Easy configuration, management, and monitoring	Yes
External keystone support	Yes
Object store features (Swift)	
Unified file and object support	Yes
High performance and High Availability	Yes
Object encryption	Yes
WAN caching with Active File Management (AFM)	Yes
Easy install, configuration, and management	Yes
Swift and S3 API Support	Yes

Table 20. Features that IBM Spectrum Scale supports for deploying the OpenStack cloud storage (continued)

Feature	Support
Support for OpenStack Swift object store features	Yes
In-place analytics with Hadoop compatibility	Yes
Object compression	Yes
Multi-region support	Yes
Policy based information life cycle management	Yes
Integrated monitoring	Yes
Large object support (5 TB)	Yes

For more information on OpenStack, see the [OpenStack Redpaper](#).

IBM Spectrum Scale product editions

IBM Spectrum Scale offers different editions that are based on functional levels.

Note: All IBM Spectrum Scale software nodes in a cluster must have the same edition installed. Capacity licensing and socket licensing cannot be mixed in the same cluster, with the exception of ESS. ESS can be mixed with IBM Spectrum Scale software in the same cluster when one of the following requirements is met:

- All nodes in the cluster run Data Access Edition or Standard Edition.
- All nodes in the cluster run Data Management Edition or Advanced Edition.

For information about capacity licensing, see “Capacity-based licensing” on page 218. For information about the features available in respective IBM Spectrum Scale editions, see [Table 21 on page 214](#).

IBM Spectrum Scale Data Access Edition

Available on AIX, Linux, and Windows. This edition provides base IBM Spectrum Scale functions. On AIX and Linux, the available features include Information Lifecycle Management (ILM), Active File Management (AFM), and Clustered NFS (CNFS). CNFS is not available on Linux on Z. On Windows, the available features include limited Information Lifecycle Management (ILM).

On Red Hat Enterprise Linux 7.x and 8.x, SLES 15, Ubuntu 20.04, and Ubuntu 22.04, the available features include the ability to enable and use the additional protocol functionality integration (NFS and SMB). On Red Hat Enterprise Linux 8.x, the object protocol functionality is also supported.

IBM Spectrum Scale Data Management Edition

Available on AIX and Linux. This edition provides all the features of the Data Access Edition and certain additional features. For more information, see [Table 21 on page 214](#).

IBM Spectrum Scale Developer Edition

Available on Red Hat Enterprise Linux on x86_64. This edition provides all the features of the Data Management Edition and it is limited to 12 TB per cluster. You can check your licensed usage on IBM Spectrum Scale Developer Edition by using the `mmlslicense --licensed-usage` command. For more information, see `mmlslicense` command in *IBM Spectrum Scale: Command and Programming Reference*.

Note: This edition is made available to the customers free of cost to try IBM Spectrum Scale features in test setups. Its use in production is not allowed. There is no support from IBM for IBM Spectrum Scale Developer Edition.

You can sign up for IBM Spectrum Scale Developer Edition [here](#).

IBM Spectrum Scale Erasure Code Edition

This edition provides identical functionality to IBM Spectrum Scale Data Management Edition plus the support for storage rich servers. IBM Spectrum Scale Erasure Code Edition provides network-dispersed erasure coding, distributing data, and metadata across the internal disks of a cluster of servers. This allows IBM Spectrum Scale to use internal disks as reliable storage with low overhead and high performance.

Legacy editions:

IBM Spectrum Scale Standard Edition

This edition provides comparable functionality to IBM Spectrum Scale Data Access Edition under socket-based licensing. Standard Edition does not provide erasure coding and it is not supported with ESS.

IBM Spectrum Scale Advanced Edition

This edition provides comparable functionality to IBM Spectrum Scale Data Management Edition under socket-based licensing. Advanced Edition does not provide erasure coding and it is not supported with ESS.

Note:

- IBM Spectrum Scale Standard Edition licenses can only be purchased by existing licensees of Standard Edition. IBM Spectrum Scale Advanced Edition licenses can only be purchased by existing licensees of Advanced Edition. All others must purchase Data Access Edition, Data Management Edition, or Erasure Code Edition.
- IBM Spectrum Scale Standard Edition and IBM Spectrum Scale Advanced Edition can be licensed by IBM Spectrum Scale Client license, IBM Spectrum Scale Server license, and IBM Spectrum Scale FPO license. For more information, see [“IBM Spectrum Scale license designation”](#) on page 215.

Table 21 on page 214 lists the availability of key features in the IBM Spectrum Scale editions.

Table 21. Features in IBM Spectrum Scale editions			
Feature	Data Access	Data Management ¹	Erasure Code Edition
Multi-protocol scalable file service with simultaneous access to a common set of data	✓	✓	✓
Facilitate data access with a global namespace, massively scalable file system, quotas and snapshots, data integrity and availability, and filesets	✓	✓	✓
Simplify management with GUI	✓	✓	✓
Improved efficiency with QoS and compression	✓	✓	✓
Create optimized tiered storage pools based on performance, locality, or cost	✓	✓	✓

Table 21. Features in IBM Spectrum Scale editions (continued)

Feature	Data Access	Data Management ¹	Erasure Code Edition
Simplify data management with Information Lifecycle Management (ILM) tools that include policy based data placement and migration	✓	✓	✓
Enable worldwide data access using AFM asynchronous replication	✓	✓	✓
Asynchronous multi-site Disaster Recovery		✓	✓
Hybrid cloud (TCT)		✓	✓
Protect data with native software encryption and secure erase, NIST compliant and FIPS certified		✓	✓
File audit logging		✓	✓
Watch folder		✓	✓
Erasure coding	ESS only	ESS only	✓

Note: ¹IBM Spectrum Scale Developer Edition provides the same features as Data Management Edition and it is limited to 12 TB per cluster.

Consult the [IBM Spectrum Scale FAQ in IBM Documentation](#) for the latest features that are included in each edition.

IBM Spectrum Scale license designation

According to the IBM Spectrum Scale Licensing Agreement, each server licensed with IBM Spectrum Scale Standard Edition or IBM Spectrum Scale Advanced Edition in the cluster must be designated as possessing an IBM Spectrum Scale Client license, an IBM Spectrum Scale File Placement Optimizer (FPO) license, or an IBM Spectrum Scale Server license.

There are three basic storage cluster models, which are storage area network (SAN), network shared disk (NSD), and shared nothing. IBM Spectrum Scale supports all three of them. For SAN and NSD clusters, a number of Server licenses are required in addition to Client licenses. For shared nothing clusters of storage rich servers, FPO licenses are required on each of those servers.

The type of license that is associated with any one server depends on the functional roles that the server is designated to perform.

IBM Spectrum Scale Client license

The IBM Spectrum Scale Client license permits exchange of data between servers that locally mount the same IBM Spectrum Scale file system. No other export of the data is permitted.

IBM Spectrum Scale Server license

The IBM Spectrum Scale Server license permits the licensed server to perform management functions, such as cluster configuration manager, quorum node, manager node, AFM gateway node, and Network Shared Disk (NSD) server. In addition, the IBM Spectrum Scale Server license permits the licensed server to share IBM Spectrum Scale data directly through any application, service protocol or method such as Network File System (NFS), Server Message Block (SMB), File Transfer

Protocol (FTP), Hypertext Transfer Protocol (HTTP), Object Protocol (OpenStack Swift, Amazon S3 API). Therefore, protocol nodes also require an IBM Spectrum Scale Server license.

IBM Spectrum Scale FPO license

The IBM Spectrum Scale FPO license permits the licensed server to perform NSD server functions for sharing IBM Spectrum Scale data with other servers that have an IBM Spectrum Scale FPO or an IBM Spectrum Scale Server license.

The full text of the Licensing Agreement is provided with the installation media and can be found at the IBM Software license agreements website (www.ibm.com/software/sla/sladb.nsf).

Use the guidance in [Figure 27](#) on [page 216](#) to decide which IBM Spectrum Scale license to buy for your requirements.

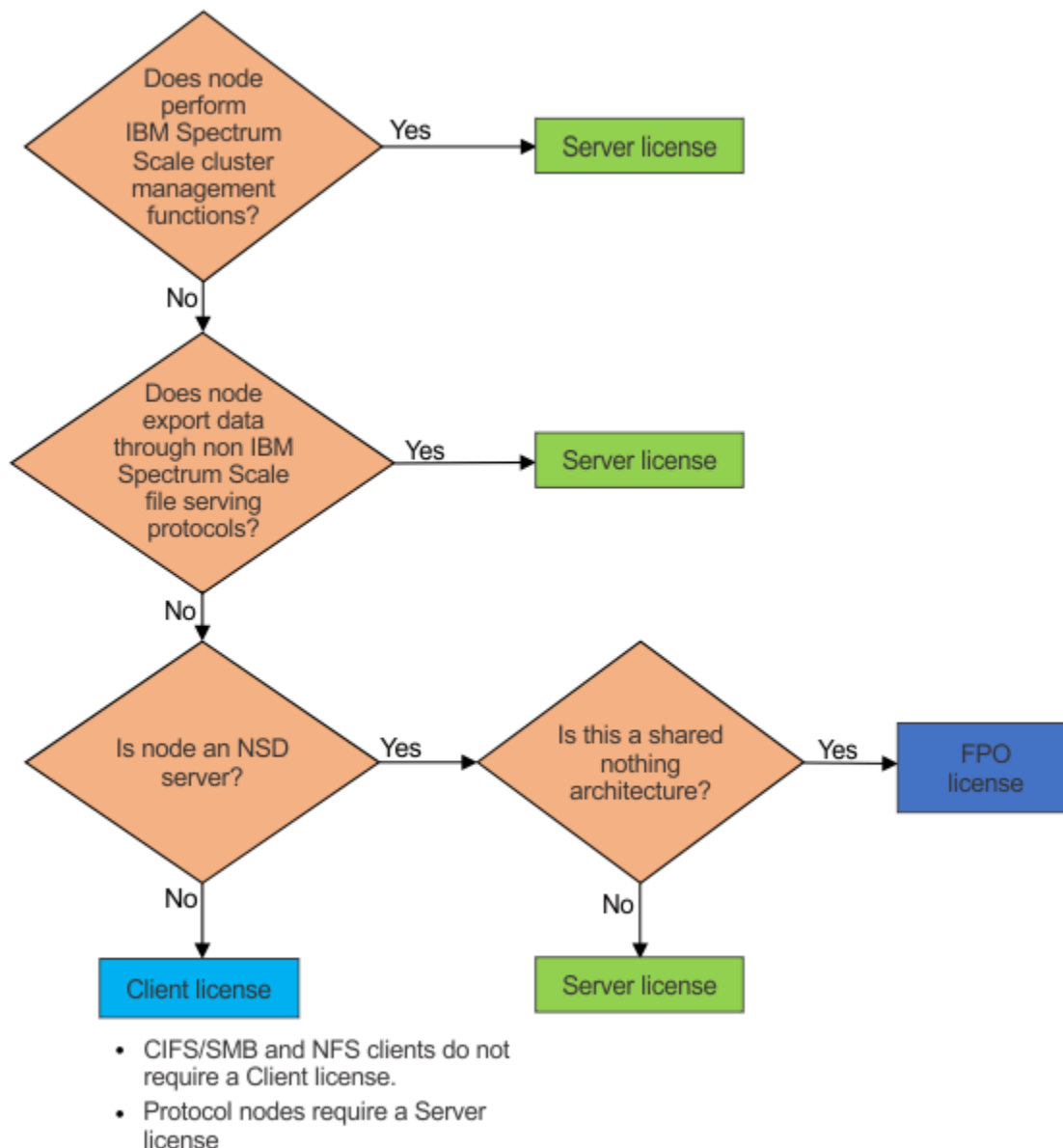


Figure 27. Guidance on which license to buy

These licenses are all valid for use in IBM Spectrum Scale Standard Edition and IBM Spectrum Scale Advanced Edition. For more information, see [“IBM Spectrum Scale product editions”](#) on [page 213](#).

With IBM Spectrum Scale Data Access Edition, IBM Spectrum Scale Data Management Edition, and IBM Spectrum Scale Erasure Code Edition, you must use capacity-based licensing. For more information, see [“Capacity-based licensing”](#) on page 218.

- You can view the number and type of licenses that are currently in effect for the cluster by using the `mmllslicense` command.
- If needed, you can change the type of license assigned to a node by using the `mmchlicense` command.

For more information, see **`mmllslicense` command** and **`mmchlicense` command** in *IBM Spectrum Scale: Command and Programming Reference*.

The following are IBM Spectrum Scale licensing considerations including considerations for using IBM Spectrum Scale with other offerings.

AFM-based Async Disaster Recovery (AFM DR) with multicluster

When using AFM-based Async Disaster Recovery (AFM DR) in a multicluster environment, both the home and the cache cluster require the IBM Spectrum Scale Advanced Edition, IBM Spectrum Scale Data Management Edition, or IBM Spectrum Scale Developer Edition.

Encryption

The encryption function that is available in IBM Spectrum Scale Advanced Edition, IBM Spectrum Scale Data Management Edition, or IBM Spectrum Scale Developer Edition requires a separate IBM Security Key Lifecycle Manager (ISKLM) license.

Encryption with multicluster

When using the IBM Spectrum Scale encryption function in a multicluster environment, each server in an IBM Spectrum Scale cluster requiring access to another cluster's encrypted file system must be licensed with IBM Spectrum Scale Advanced Edition (Client, Server, or FPO as appropriate) or IBM Spectrum Scale Data Management Edition.

Hadoop access

The IBM Spectrum Scale Hadoop connector can be used with all license types (Client, Server, FPO) and all Editions (Standard, Advanced, and Data Management). There is no additional license requirement for Hadoop access. A Hadoop node using the connector needs no IBM Spectrum Scale license. The Hadoop node can connect to a node in the IBM Spectrum Scale cluster, which is licensed as Server because it is exporting data, and access the file system directly by using the Hadoop connector.

ILMT service

The IBM License Metric Tool (ILMT) ensures efficiency in inventory deduction and better estimation of license consumption. To use the ILMT service, you must schedule the command **`mmllslicense --ilmt-data`** to run once a week, by using mechanisms like cron. For more information, see *mmllslicense* in the *IBM Spectrum Scale: Command and Programming Reference*

IBM Spectrum Scale with IBM Spectrum Protect

When using IBM Spectrum Scale with IBM Spectrum Protect for backup and restore, each IBM Spectrum Scale node performing data movement requires an IBM Spectrum Protect license.

IBM Spectrum Scale with IBM Spectrum Protect for Space Management

When using IBM Spectrum Scale with IBM Spectrum Protect for Space Management for migration and recall, each IBM Spectrum Scale node performing data movement requires an IBM Spectrum Protect for Space Management license.

IBM Spectrum Scale with IBM Spectrum Archive Enterprise Edition (EE)

In addition to an IBM Spectrum Archive Enterprise Edition (EE) license, a server in the IBM Spectrum Scale cluster which is being used for IBM Spectrum Archive Enterprise Edition (EE) requires IBM Spectrum Scale to be installed with the correct license (ordered separately). The IBM Spectrum Archive Enterprise Edition (EE) server requires an IBM Spectrum Scale Server license, if the IBM Spectrum Archive Enterprise Edition (EE) server is also being used for any of the functions requiring

an IBM Spectrum Scale Server license, such as NSD server, quorum, management, etc. Otherwise, the IBM Spectrum Archive Enterprise Edition (EE) server can be licensed with an IBM Spectrum Scale client license. The IBM Spectrum Archive Enterprise Edition (EE) server's function of moving data from the file system to a tape drive is not considered as a type of data export that requires an IBM Spectrum Scale server license.

Virtualization licensing

In an IBM Spectrum Scale environment containing VMs or LPARs, licenses are required for the sockets available to the IBM Spectrum Scale VMs or LPARs on a physical server. The number of licenses required is the number of cores available to IBM Spectrum Scale. For more information, see *Virtualization questions* in [IBM Spectrum Scale FAQ](#) in IBM Documentation.

Capacity-based licensing

You can use capacity-based licensing to license IBM Spectrum Scale based on the storage capacity being managed in an IBM Spectrum Scale cluster.

The storage capacity to be licensed is the capacity in Terabytes (TiB) from all NSDs in the IBM Spectrum Scale cluster. The capacity to be licensed is not affected by using functions such as replication or compression or by doing tasks such as creating or deleting files, file systems, or snapshots.

For example, if you have 100 TiB of capacity from all NSDs in the IBM Spectrum Scale cluster and if you have set the replication factor to 2, for 50 TiB of application data, 100 TiB of disk capacity is consumed. In this case, the capacity to be licensed is 100 TiB, not 50 TiB.

The capacity to be licensed changes only when an NSD is added or deleted from the IBM Spectrum Scale cluster.

You can view the cluster capacity for licensing by using one of the following methods:

- Select the **About** option from the menu that is available in the upper right corner of the IBM Spectrum Scale management GUI.
- Issue the **mmlslicense --capacity** command.

Capacity-based licensing can be used only with IBM Spectrum Scale Data Access Edition, IBM Spectrum Scale Data Management Edition, and IBM Spectrum Scale Erasure Code Edition. The Data Access Edition provides identical functionality as the Standard Edition. The Data Management Edition provides identical functionality as the Advanced Edition. The Erasure Code Edition provides identical functionality as the Data Management Edition plus support for storage rich servers.

Capacity-based licensing can also be used to license an Elastic Storage Server (ESS) system. For ESS, the capacity to be licensed is the capacity after applying IBM Spectrum Scale RAID. The exact capacity to be licensed depends on the RAID code selected, 8+2p or 8+3p.

Chapter 2. Planning for IBM Spectrum Scale

Planning for GPFS

Although you can modify your GPFS configuration after it has been set, a little consideration before installation and initial setup will reward you with a more efficient and immediately useful file system.

During configuration, GPFS requires you to specify several operational parameters that reflect your hardware resources and operating environment. During file system creation, you can specify parameters that are based on the expected size of the files or you can let the default values take effect.

The installation toolkit is also available to assist with GPFS installation on Linux nodes. For more information, see [“Overview of the installation toolkit” on page 386](#)

Hardware requirements

You can validate that your hardware meets GPFS requirements by taking the steps outlined in this topic.

1. Consult the [IBM Spectrum Scale FAQ in IBM Documentation](#) for the latest list of:

- Supported server hardware
- Tested disk configurations
- Maximum cluster size
- Additional hardware requirements for protocols

2. Provide enough disks to contain the file system. Disks can be:

- SAN-attached to each node in the cluster
- Attached to one or more NSD servers
- A mixture of directly-attached disks and disks that are attached to NSD servers

For more information, see the *Network Shared Disk (NSD) creation considerations* topic in *IBM Spectrum Scale: Administration Guide*.

3. When doing network-based NSD I/O, GPFS passes a large amount of data between its daemons. For NSD server-to-client traffic, it is suggested that you configure a dedicated high-speed network solely for GPFS communications when the following are true:

- There are NSD disks configured with servers providing remote disk capability.
- Multiple GPFS clusters are sharing data using NSD network I/O.

For more information, see the *Disk considerations* topic in *IBM Spectrum Scale: Administration Guide*.

GPFS communications require static IP addresses for each GPFS node. IP address takeover operations that transfer the address to another computer are not allowed for the GPFS network. Other IP addresses within the same computer that are not used by GPFS can participate in IP takeover. To provide availability or additional performance, GPFS can use virtual IP addresses created by aggregating several network adapters using techniques such as EtherChannel or channel bonding.

Cluster Export Services (CES) have dedicated network addresses to support the SMB, NFS, HDFS, and object protocols. These network addresses are appropriately assigned to CES nodes and they facilitate CES node failover and failback operations. File and object clients use the public IPs to access data on GPFS™ file systems.

For additional information, see *CES network configuration* in *IBM Spectrum Scale: Administration Guide*.

Software requirements

IBM Spectrum Scale planning includes understanding the latest software requirements.

Note: This topic lists the additional software requirements to use specific functions after successful installation. It does not list the dependencies for installing IBM Spectrum Scale packages.

- GPFS is supported on AIX, Linux, and Windows.
- Kernel development files and compiler utilities are required to build the GPFS portability layer on Linux nodes. The required RPMs or packages for each supported Linux distribution are as follows:
 - **Red Hat Enterprise Linux RPMs**
 - kernel-devel
 - cpp
 - gcc
 - gcc-c++
 - binutils
 - elfutils-libelf-devel (Required for Red Hat Enterprise Linux 8.x)
 - **SLES Linux RPMs**
 - kernel-default-devel
 - cpp
 - gcc
 - gcc-c++
 - binutils
 - libelf-devel
 - **Ubuntu Linux packages**
 - linux-generic
 - cpp
 - gcc
 - g++
 - binutils
 - libelf-dev

Software requirements for specific IBM Spectrum Scale functions

Additional software requirements to use specific functions after successful installation are as follows.

Required package for active file management (AFM)

nfs-utils

Required packages for Cluster Export Services (CES) and CNFS

- ethtool
- nfs-utils (on Ubuntu: nfs-common)
- rpcbind
- psmisc
- iputils (on Ubuntu: iputils-arping)
- ifupdown (only on Ubuntu; there is no additional package requirement for RHEL/SLES)

Note: On a Linux node that is not a protocol node running the NFS service, it is recommended to disable the port mapper service (also called rpcbind).

Required package for SLES 15 CES nodes

`gdb`

Requirements for file authentication

Before configuring file authentication, ensure that prerequisites are met to avoid installation failure. For more information, see the topic *Configuring authentication and ID mapping for file access* in *IBM Spectrum Scale: Administration Guide*.

You can use Active Directory (AD) authentication. For more information, see the topic *Integrating with AD server* in *IBM Spectrum Scale: Administration Guide*.

For more information, see the *Integrating with LDAP server* topic in *IBM Spectrum Scale: Administration Guide*.

Required packages as per OS and Authentication schemes

```
[ad:RHEL]="bind-utils"
[ad:SLES]="bind-utils libarchive13"
[ad:UBUNTU]="dnsutils"

[ldap:RHEL]="openldap-clients sssd sssd-tools"
[ldap:SLES]="openldap2-client sssd sssd-tools sssd-ldap"
[ldap:UBUNTU]="ldap-utils sssd sssd-tools"

[nis:RHEL]="sssd ybind sssd-proxy"
[nis:SLES]="sssd ybind sssd-proxy"
[nis:UBUNTU]="sssd nis sssd-proxy"

[kerberos:RHEL]="krb5-workstation"
[kerberos:SLES]="krb5-client sssd-krb5"
[kerberos:UBUNTU]="krb5-user"
```

Required packages for performance monitoring

- `boost-regex` on Red Hat Enterprise Linux
- `libboost_regex` on SLES
- `libboost-regex-dev` on Ubuntu Linux

Important: Ensure that these packages are installed on the system before the installation of the performance monitoring tool.

Required package for the IBM Spectrum Scale monitoring service on AIX and Linux

- Python 3

Note:

- On Linux, Python 3 is usually installed automatically.
- Python 3 is shipped with AIX 7.3.

On AIX 7.2, you must manually install Python 3.0 or later. For more information, see [AIX Toolbox for Linux Applications](#). To manually install, one of the ways is to configure Yum with the AIX toolbox. For more information, see [Configuring YUM and creating local repositories on IBM AIX](#).

- On Windows, manually install Python 3 under the Cygwin environment as described in the following steps. Windows native (non-Cygwin) distributions of Python 3 are not supported.
 1. From <http://www.cygwin.com>, download and run the Cygwin 64-bit setup program `setup-x86_64.exe`.
 2. In the "Select Packages" window, click **View > Category**.
 3. Click **All > Python > Python3** and select the latest level.
 4. Follow the instructions to complete the installation.

Requirements for the management GUI

The following packages must be installed on the node on which GUI needs to be installed:

- PostgreSQL server including contrib package. The PostgreSQL server must be specific to the Linux distribution.
- IBM Spectrum Scale Java™ Runtime Environment (JRE) (gpfs.java RPM).
- IBM Spectrum Scale performance collector (gpfs.pmcollector RPM).
- openssl for HTTPS certificate management.
- gpfs.gskit to use CCR.

Prerequisites and required packages for IBM Spectrum Scale for object storage

To install the object protocol, pre-configuration of OpenStack repositories is required on all protocol nodes. For more information, see [“OpenStack repository configuration required by the object protocol”](#) on page 319.

When the IBM Spectrum Scale object protocol is installed, the following SELinux packages are also installed:

- selinux-policy-base at 3.13.1-23 or higher
- selinux-policy-targeted at 3.12.1-153 or higher



Attention:

- The object protocol is not supported in IBM Spectrum Scale 5.1.0.0. If you want to deploy object, install the IBM Spectrum Scale 5.1.0.1 or a later release.
- If SELinux is disabled during installation of IBM Spectrum Scale for object storage, enabling SELinux after installation is not supported.

Required packages for clustered watch folder

Every node that is mounting the watched file system must have the following packages installed:

- For RHEL, the librdkafka package requires the openssl-devel and cyrus-sasl-devel packages.
- For Ubuntu, the librdkafka package requires the libssl-dev and libsasl2-dev packages.
- librdkafka (gpfs.librdkafka rpm/package).

For more information, see [“Requirements, limitations, and support for clustered watch folder”](#) on page 491.

Required package to use the mmchconfig numaMemoryInterleave parameter numactl

Required package to use the mmhealth command on Ubuntu Linux sqlite3

Required packages for mmprotocoltrace

Some of the advanced tracing components of the **mmprotocoltrace** command require specific packages to be installed on all nodes that need to participate in tracing related operations.

- To enable network tracing with **mmprotocoltrace**, you need to install the tcpdump package.
- To enable the syscalls-tracing for SMB, you need to install the strace package.

Prerequisites for the installation toolkit

For information on prerequisites for using the installation toolkit, see [“Preparing to use the installation toolkit”](#) on page 404.

For additional prerequisites on Linux, see [“Installation prerequisites”](#) on page 352.

Consult the [IBM Spectrum Scale FAQ](#) in IBM Documentation for the latest list of:

- AIX environments
- Linux distributions

- Linux kernel versions
- Windows environments

Recoverability considerations

Good file system planning requires several decisions about recoverability. After you make these decisions, GPFS parameters enable you to create a highly-available file system with rapid recovery from failures.

- At the disk level, consider preparing disks for use with your file system by specifying failure groups that are associated with each disk. With this configuration, information is not vulnerable to a single point of failure. See [“Network Shared Disk \(NSD\) creation considerations” on page 239](#).
- At the file system level, consider replication through the metadata and data replication parameters. See [“File system replication parameters” on page 263](#).

Additionally, GPFS provides several layers of protection against failures of various types.

Node failure

If a node fails, GPFS prevents the continuation of I/O from the failing node and replays the file system metadata log for the failed node.

GPFS prevents the continuation of I/O from a failing node through a GPFS-specific fencing mechanism called *disk leasing*. When a node has access to file systems, it obtains disk leases that allow it to submit I/O. However, when a node fails, that node cannot obtain or renew a disk lease. When GPFS selects another node to perform recovery for the failing node, it first waits until the disk lease for the failing node expires. This allows for the completion of previously submitted I/O and provides for a consistent file system metadata log. Waiting for the disk lease to expire also avoids data corruption in the subsequent recovery step.

To reduce the amount of time it takes for disk leases to expire, you can use Persistent Reserve (SCSI-3 protocol). If Persistent Reserve (configuration parameter: `usePersistentReserve`) is enabled, GPFS prevents the continuation of I/O from a failing node by fencing the failed node that uses a feature of the disk subsystem called Persistent Reserve. Persistent Reserve allows the failing node to recover faster because GPFS does not need to wait for the disk lease on the failing node to expire. For additional information, refer to [“Reduced recovery time by using Persistent Reserve” on page 228](#). For further information about recovery from node failure, see *Installation and configuration issues in IBM Spectrum Scale: Problem Determination Guide*.

There is a temporary impact to I/O during file system recovery from node failure. Recovery involves rebuilding metadata structures that might be under modification at the time of the failure. If the failing node is acting as the file system manager when it fails, the delay is longer and proportional to the level of activity on the file system at the time of failure. In this case, the failover file system management task happens automatically to a surviving node.

Managing node failures also involves sizing the solution adequately so that remaining nodes in the cluster can support a node down situation such as a planned system maintenance or an unplanned node failure in terms of bandwidth and throughput. For protocols, this includes supporting SMB, NFS, or Object connections that have to fail over to another CES node in the cluster if a node fails.

Quorum

GPFS uses a cluster mechanism called quorum to maintain data consistency in the event of a node failure.

Quorum operates on the principle of majority rule. This means that a majority of the nodes in the cluster must be successfully communicating before any node can mount and access a file system. This keeps any nodes that are cut off from the cluster (for example, by a network failure) from writing data to the file system.

During node failure situations, quorum needs to be maintained in order for the cluster to remain online. If quorum is not maintained due to node failure, GPFS unmounts local file systems on the remaining nodes and attempts to reestablish quorum, at which point file system recovery occurs. For this reason it

is important that the set of quorum nodes be carefully considered (refer to [“Selecting quorum nodes”](#) on page 225 for additional information).

GPFS quorum must be maintained within the cluster for GPFS to remain active. If the quorum semantics are broken, GPFS performs recovery in an attempt to achieve quorum again. GPFS can use one of two methods for determining quorum:

- Node quorum
- Node quorum with tiebreaker disks

Node quorum

Node quorum is the default quorum algorithm for GPFS.

With node quorum:

- Quorum is defined as one plus half of the *explicitly defined* quorum nodes in the GPFS cluster.
- There are no default quorum nodes; you must specify which nodes have this role.

For example, in Figure 28 on page 224, there are three quorum nodes. In this configuration, GPFS remains active as long as there are two quorum nodes available.

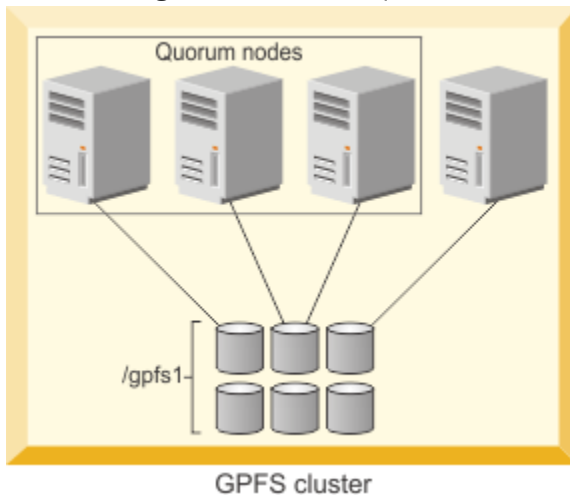


Figure 28. GPFS configuration using node quorum

Node quorum with tiebreaker disks

When running on small GPFS clusters, you might want to have the cluster remain online with only one surviving node.

To achieve this, you need to add a tiebreaker disk to the quorum configuration. Node quorum with tiebreaker disks allows you to run with as little as one quorum node available as long as you have access to a majority of the quorum disks (refer to Figure 29 on page 225). Enabling node quorum with tiebreaker disks starts by designating one or more nodes as quorum nodes. Then one to three disks are defined as tiebreaker disks using the **tiebreakerDisks** parameter on the `mmchconfig` command. You can designate any disk to be a tiebreaker.

When utilizing node quorum with tiebreaker disks, there are specific rules for cluster nodes and for tiebreaker disks.

Cluster node rules:

1. There is a maximum of eight quorum nodes.
2. All quorum nodes need to have access to all of the tiebreaker disks.
3. You may have an unlimited number of non-quorum nodes.
4. If a network connection fails, which causes the loss of quorum, and quorum is maintained by tiebreaker disks, the following rationale is used to re-establish quorum. If a group has the cluster manager, it is the "survivor". The cluster manager can give up its role if it communicates with fewer

than the minimum number of quorum nodes as defined by the **minQuorumNodes** configuration parameter. In this case, other groups with the minimum number of quorum nodes (if they exist) can choose a new cluster manager.

Changing quorum semantics:

When using the cluster configuration repository (CCR) to store configuration files, the total number of quorum nodes is limited to eight, regardless of quorum semantics, but the use of tiebreaker disks can be enabled or disabled at any time by issuing an `mmchconfig tiebreakerDisks` command. The change will take effect immediately, and it is not necessary to shut down GPFS when making this change.

Tiebreaker disk rules:

- You can have one, two, or three tiebreaker disks. However, you should use an odd number of tiebreaker disks.
- Among the quorum node groups that appear after an interconnect failure, only those having access to a majority of tiebreaker disks can be candidates to be the survivor group.
- Tiebreaker disks must be connected to all quorum nodes.
- In a CCR-based cluster, when adding tiebreaker disks:
 - GPFS should be up and running, if tiebreaker disks are part of the file system.
 - GPFS can be either running or shut down, if tiebreaker disks are not a part of the file system.

In [Figure 29 on page 225](#) GPFS remains active with the minimum of a single available quorum node and two available tiebreaker disks.

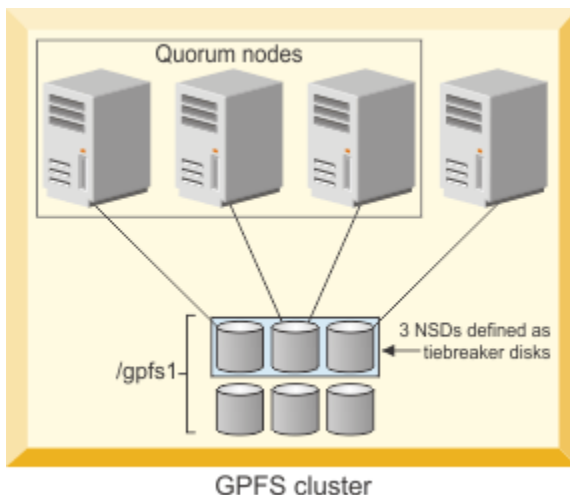


Figure 29. GPFS configuration using node quorum with tiebreaker disks

When a quorum node detects loss of network connectivity, but before GPFS runs the algorithm that decides if the node will remain in the cluster, the `tiebreakerCheck` event is triggered. This event is generated only in configurations that use quorum nodes with tiebreaker disks. It is also triggered on the cluster manager periodically by a challenge-response thread to verify that the node can still continue as cluster manager.

Selecting quorum nodes

To configure a system with efficient quorum nodes, follow these rules.

- Select nodes that are likely to remain active
 - If a node is likely to be rebooted or require maintenance, do not select that node as a quorum node.
- Select nodes that have different failure points such as:
 - Nodes that are located in different racks
 - Nodes that are connected to different power panels
- You should select nodes that GPFS administrative and serving functions rely on such as:

- Network Shared Disk (NSD) servers
- Select an odd number of nodes as quorum nodes
 - The maximum is eight quorum nodes.
- Having a large number of quorum nodes might increase the time required for startup and failure recovery.
 - Having more than seven quorum nodes does not guarantee higher availability.
 - All quorum nodes must have access to all of the tiebreaker disks.

The `/var/mmfs` directory on quorum nodes cannot be on `tmpfs` or `ramfs` file system.

Note: When adding a quorum node by using either `mmcrcluster` or `mmchnode` or when you enable CCR from server-based cluster configuration method, then the system checks whether the `/var/mmfs` directory on the designated quorum node is not on `tmpfs` or `ramfs`. If `/var/mmfs` is from `tmpfs` or `ramfs`, the command output prints an error message.

Network Shared Disk server and disk failure

The three most common reasons why data becomes unavailable are disk failure, disk server failure with no redundancy, and failure of a path to the disk.

In the event of a disk failure in which GPFS can no longer read or write to the disk, GPFS will discontinue use of the disk until it returns to an available state. You can guard against loss of data availability from disk failure by:

- Utilizing hardware data protection as provided by a Redundant Array of Independent Disks (RAID) device (see [Figure 30 on page 226](#))

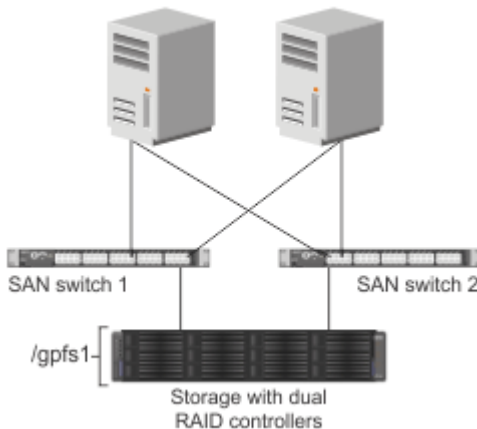


Figure 30. An example of a highly available SAN configuration for a GPFS file system

- Utilizing the GPFS data and metadata replication features (see [“Increased data availability” on page 2](#)) along with the designation of failure groups (see [“Network Shared Disk \(NSD\) creation considerations” on page 239](#) and [Figure 31 on page 227](#))

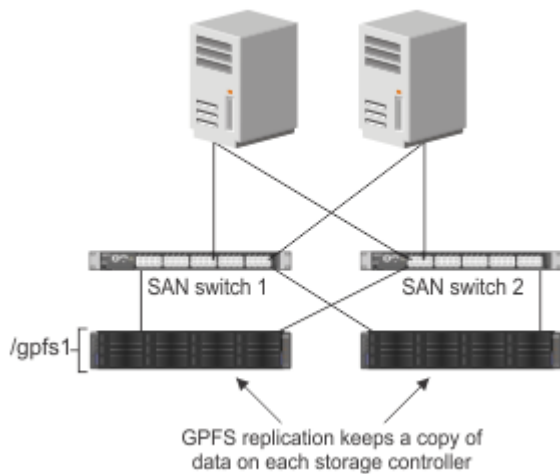


Figure 31. Configuration using GPFS replication for improved availability

It is suggested that you consider RAID as the first level of redundancy for your data and add GPFS replication if you desire additional protection.

In the event of an NSD server failure in which a GPFS client can no longer contact the node that provides remote access to a disk, GPFS discontinues the use of the disk. You can guard against loss of an NSD server availability by using common disk connectivity on multiple NSD server nodes and specifying multiple Network Shared Disk servers for each common disk.

Note: In the event that a path to a disk fails, GPFS reports a disk failure and marks the disk down. To bring the disk back online, first follow the directions supplied by your storage vendor to determine and repair the failure.

Guarding against loss of data availability due to path failure

You can guard against loss of data availability from failure of a path to a disk by doing the following:

- Creating multiple NSD servers for each disk

As GPFS determines the available connections to disks in the file system, it is recommended that you always define more than one NSD server for each disk. GPFS allows you to define up to eight NSD servers for each NSD. In a SAN configuration where NSD servers have also been defined, if the physical connection is broken, GPFS dynamically switches to the next available NSD server (as defined on the server list) and continues to provide data. When GPFS discovers that the path has been repaired, it moves back to local disk access. This is the default behavior, which can be changed by designating file system mount options. For example, if you never want a node to use the NSD server path to a disk, even if the local path fails, you can set the `-o useNSDserver` mount option to `never`. You can set the mount option using the `mmchfs`, `mmmount`, `mmremotefs`, and `mount` commands.

Important: In Linux on Z, it is mandatory to have multiple paths to one SCSI disk (LUN) to avoid single path of failure. The coalescing of the paths to one disk is done by the kernel (via the device-mapper component). As soon as the paths are coalesced, a new logical, multipathed device is created, which is used for any further (administration) task. (The single paths can no longer be used.)

The multipath device interface name depends on the distribution and is configurable:

SUSE

`/dev/mapper/Unique_WW_Identifier`

For example: `/dev/mapper/36005076303ffc56200000000000010cc`

Red Hat

`/dev/mapper/mpath*`

To obtain information about a multipathed device, use the `multipath` tool as shown in the following example:

```
# multipath -ll
```

The system displays output similar to this:

```
36005076303ffc562000000000000010cc dm-0 IBM,2107900  
[size=5.0G][features=1 queue_if_no_path][hwhandler=0]  
\_ round-robin 0 [prio=2][active]  
  \_ 1:0:0:0 sdb 8:16 [active][ready]  
  \_ 0:0:0:0 sda 8:0 [active][ready]
```

See the question, "What considerations are there when setting up DM-MP multipath service" in the [IBM Spectrum Scale FAQ in IBM Documentation](#).

- Using an I/O driver that provides multiple paths to the disks for failover purposes

Failover is a path-management algorithm that improves the reliability and availability of a device because the system automatically detects when one I/O path fails and reroutes I/O through an alternate path.

Reduced recovery time by using Persistent Reserve

Persistent Reserve (PR) provides a mechanism for reducing recovery times from node failures.

To enable PR and to obtain recovery performance improvements, your cluster requires a specific environment:

- All disks must be PR-capable. For a list of devices supported with PR, see [this question in IBM Spectrum Scale FAQ in IBM Documentation](#).
- On AIX, all disks must be hdisks. Starting with 3.5.0.16, it is also possible to use a logical volume as a descOnly disk without disabling the use of Persistent Reserve. For more information, see the [IBM Spectrum Scale FAQ in IBM Documentation](#).

On Linux, typically all disks must be generic (/dev/sd*) or DM-MP (/dev/dm-*) disks.

However, for Linux on Z, multipath device names are required for SCSI disks, and the names depend on the distribution and are configurable. For more information, see [“Guarding against loss of data availability due to path failure” on page 227](#).

- If the disks have NSD servers that are defined, all NSD server nodes must be running the same operating system (AIX or Linux).
- If the disks are SAN-attached to all nodes, all nodes in the cluster must be running the same operating system (AIX or Linux).

For quicker recovery times when you use PR, set the **failureDetectionTime** configuration parameter on the **mmchconfig** command. For example, for quick recovery a recommended value would be 10:
mmchconfig failureDetectionTime=10

You must explicitly enable PR by specifying the **usePersistentReserve** parameter on the **mmchconfig** command. If you set **usePersistentReserve=yes**, GPFS attempts to set up PR on all of the PR capable disks. All subsequent NSDs are created with PR enabled if they are PR capable. However, PR is only supported in the home cluster. Therefore, access to PR-enabled disks from another cluster must be through an NSD server that is in the home cluster and not directly to the disk (for example, through a SAN).

GPFS cluster creation considerations

A GPFS cluster is created using the **mmcrcluster** command. On supported Linux distributions, you can use the installation toolkit to create a GPFS cluster. For more information, see [“Overview of the installation toolkit” on page 386](#).

[Table 22 on page 229](#) details the cluster creation options, how to change the options, and the default values for each option.

Table 22. GPFS cluster creation options

Options	Command to change the option	Default value
“Creating an IBM Spectrum Scale cluster” on page 230	mmaddnode mmdelnode	None
Node designation: manager or client. See “Creating an IBM Spectrum Scale cluster” on page 230 .	mmchnode	client
Node designation: quorum or nonquorum. See “Creating an IBM Spectrum Scale cluster” on page 230 .	mmchnode	nonquorum
“Remote shell command” on page 233	mmchcluster	/usr/bin/ssh
“Remote file copy command” on page 234	mmchcluster	/usr/bin/scp
“Cluster name” on page 235	mmchcluster	The node name of the primary GPFS cluster configuration server.
Cluster configuration information. See “IBM Spectrum Scale cluster configuration information” on page 232 .	mmchcluster	The Cluster Configuration Repository (CCR) is enabled.
GPFS administration adapter port name. See “GPFS node adapter interface names” on page 229	mmchnode	Same as the GPFS communications adapter port name.
GPFS communications adapter port name. See “GPFS node adapter interface names” on page 229 .	mmchnode	None
“User ID domain for the cluster” on page 235 .	mmchconfig	The name of the GPFS cluster.
“Starting GPFS automatically” on page 236	mmchconfig	no
“Cluster configuration file” on page 237	Not applicable	None

GPFS node adapter interface names

An adapter interface name refers to the hostname or IP address that GPFS uses to communicate with a node. Specifically, the hostname or IP address identifies the communications adapter over which the GPFS daemons or GPFS administration commands communicate.

The administrator can specify two node adapter interface names for each node in the cluster:

GPFS node name

Specifies the name of the node adapter interface to be used by the GPFS daemons for internode communication.

GPFS admin node name

Specifies the name of the node adapter interface to be used by GPFS administration commands when communicating between nodes. If not specified, the GPFS administration commands use the same node adapter interface used by the GPFS daemons.

These names can be specified by means of the node descriptors passed to the `mmaddnode` or `mmcrcluster` command and can later be changed with the `mmchnode` command.

If multiple adapters are available on a node, this information can be communicated to GPFS by means of the `subnets` parameter on the `mmchconfig` command.

Related concepts

Creating an IBM Spectrum Scale cluster

When you create an IBM Spectrum Scale cluster, you can either create a small cluster of one or more nodes and later add nodes to it or you can create a cluster with all its nodes in one step.

IBM Spectrum Scale cluster configuration information

You can use the Cluster Configuration Repository (CCR) to maintain cluster configuration information.

Remote shell command

GPFS commands need to be able to communicate across all nodes in the cluster. To achieve this, the GPFS commands use the remote shell command that you specify on the `mmcrcluster` command or the `mmchcluster` command.

Remote file copy command

The GPFS commands must maintain a number of configuration files across all nodes in the cluster. To achieve this, the GPFS commands use the remote file copy command that you specify on the `mmcrcluster` command or the `mmchcluster` command.

Cluster name

Provide a name for the cluster by issuing the `-C` option on the `mmcrcluster` command.

User ID domain for the cluster

This option is the user ID domain for a cluster when accessing a file system remotely.

Cluster configuration file

GPFS provides default configuration values, so a cluster configuration file is not required to create a cluster.

Related tasks

Starting GPFS automatically

You can specify whether to start the GPFS daemon automatically on a node when it is started.

Creating an IBM Spectrum Scale cluster

When you create an IBM Spectrum Scale cluster, you can either create a small cluster of one or more nodes and later add nodes to it or you can create a cluster with all its nodes in one step.

Two methods are available for creating an IBM Spectrum Scale cluster:

- Run the `mmcrcluster` command to create a cluster that contains one or more nodes, and later run the `mmaddnode` command as needed to add nodes to the cluster. This method is more flexible and is appropriate when you want to build a cluster step by step.
- Run the `mmcrcluster` command to create a cluster and at the same time to add a set of nodes to the cluster. This method is quicker when you already know which nodes you want to add to the cluster.

Whichever method you choose, you can later add nodes to the cluster with the `mmaddnode` command or delete nodes with the `mmdelnode` command. For more information, see the topics *mmcrcluster command* and *mmaddnode command* in the *IBM Spectrum Scale: Command and Programming Reference* guide.

With both commands, you specify node descriptors to identify the nodes that you are adding to the cluster. You can list node descriptors either on the command line or in a separate node descriptor file. A node descriptor has the following format:

```
NodeName:NodeDesignations:AdminNodeName
```

NodeName is a required parameter. *NodeDesignations* and *AdminNodeName* are optional parameters.

NodeName

The host name or IP address of the node for GPFS daemon-to-daemon communication.

The host name or IP address that is used for a node must refer to the communication adapter over which the GPFS daemons communicate. Alias names are not allowed. You can specify an IP address at NSD creation, but it will be converted to a host name that must match the GPFS node name. You can specify a node using any of these forms:

- Short hostname (for example, h135n01)
- Long hostname (for example, h135n01.frf.ibm.com)
- IP address (for example, 7.111.12.102)

Note: Host names should always include at least one alpha character, and they should not start with a hyphen (-).

Whichever form you specify, the other two forms must be defined correctly in DNS or the hosts file.

NodeDesignations

An optional, "-" separated list of node roles.

- `manager | client` – Indicates whether a node is part of the node pool from which file system managers and token managers can be selected. The default is `client`, which means do not include the node in the pool of manager nodes. For detailed information on the manager node functions, see [“The file system manager” on page 9](#).

In general, it is a good idea to define more than one node as a manager node. How many nodes you designate as manager depends on the workload and the number of GPFS server licenses you have. If you are running large parallel jobs, you may need more manager nodes than in a four-node cluster supporting a Web application. As a guide, in a large system there should be a different file system manager node for each GPFS file system.

- `quorum | nonquorum` – This designation specifies whether or not the node should be included in the pool of nodes from which quorum is derived. The default is `nonquorum`. You need to designate at least one node as a quorum node. It is recommended that you designate at least the primary and secondary cluster configuration servers and NSD servers as quorum nodes.

How many quorum nodes you designate depends upon whether you use node quorum or node quorum with tiebreaker disks. See [“Quorum” on page 223](#).

AdminNodeName

Specifies an optional field that consists of a node name to be used by the administration commands to communicate between nodes.

If *AdminNodeName* is not specified, the *NodeName* value is used.

Follow these rules when you add nodes to an IBM Spectrum Scale cluster:

- While a node may mount file systems from multiple clusters, the node itself may only reside in a single cluster. Nodes are added to a cluster using the `mmcrcluster` or `mmaddnode` command.
- The nodes must be available when they are added to a cluster. 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 then issue the `mmaddnode` command to add those nodes.
- Designate at least one but not more than seven nodes as quorum nodes. When not using tiebreaker disks, you can designate more quorum nodes, but it is recommended to use fewer than eight if possible. When using server-based configuration repository, it is recommended that you designate the cluster configuration servers as quorum nodes. How many quorum nodes altogether you will have depends on whether you intend to use the node quorum with tiebreaker algorithm or the regular node based quorum algorithm. For more details, see [“Quorum” on page 223](#).

Related concepts

GPFS node adapter interface names

An adapter interface name refers to the hostname or IP address that GPFS uses to communicate with a node. Specifically, the hostname or IP address identifies the communications adapter over which the GPFS daemons or GPFS administration commands communicate.

IBM Spectrum Scale cluster configuration information

You can use the Cluster Configuration Repository (CCR) to maintain cluster configuration information.

Remote shell command

GPFS commands need to be able to communicate across all nodes in the cluster. To achieve this, the GPFS commands use the remote shell command that you specify on the `mmcrcluster` command or the `mmchcluster` command.

Remote file copy command

The GPFS commands must maintain a number of configuration files across all nodes in the cluster. To achieve this, the GPFS commands use the remote file copy command that you specify on the `mmcrcluster` command or the `mmchcluster` command.

Cluster name

Provide a name for the cluster by issuing the `-C` option on the `mmcrcluster` command.

User ID domain for the cluster

This option is the user ID domain for a cluster when accessing a file system remotely.

Cluster configuration file

GPFS provides default configuration values, so a cluster configuration file is not required to create a cluster.

Related tasks

Starting GPFS automatically

You can specify whether to start the GPFS daemon automatically on a node when it is started.

IBM Spectrum Scale cluster configuration information

You can use the Cluster Configuration Repository (CCR) to maintain cluster configuration information.

It is a good practice to use the Cluster Configuration Repository (CCR) to maintain the cluster configuration information. By default, the `mmcrcluster` command enables the CCR. For more information, see [“Cluster configuration repository” on page 26](#).



Attention: The primary and secondary configuration server functionality is deprecated. For more information, see the notes in the topics *mmcrcluster command* and *mmchcluster command* in the *IBM Spectrum Scale: Command and Programming Reference* guide.

Related concepts

GPFS node adapter interface names

An adapter interface name refers to the hostname or IP address that GPFS uses to communicate with a node. Specifically, the hostname or IP address identifies the communications adapter over which the GPFS daemons or GPFS administration commands communicate.

Creating an IBM Spectrum Scale cluster

When you create an IBM Spectrum Scale cluster, you can either create a small cluster of one or more nodes and later add nodes to it or you can create a cluster with all its nodes in one step.

Remote shell command

GPFS commands need to be able to communicate across all nodes in the cluster. To achieve this, the GPFS commands use the remote shell command that you specify on the `mmcrcluster` command or the `mmchcluster` command.

Remote file copy command

The GPFS commands must maintain a number of configuration files across all nodes in the cluster. To achieve this, the GPFS commands use the remote file copy command that you specify on the `mmcrcluster` command or the `mmchcluster` command.

Cluster name

Provide a name for the cluster by issuing the `-C` option on the `mmcrcluster` command.

User ID domain for the cluster

This option is the user ID domain for a cluster when accessing a file system remotely.

Cluster configuration file

GPFS provides default configuration values, so a cluster configuration file is not required to create a cluster.

Related tasks

Starting GPFS automatically

You can specify whether to start the GPFS daemon automatically on a node when it is started.

Remote shell command

GPFS commands need to be able to communicate across all nodes in the cluster. To achieve this, the GPFS commands use the remote shell command that you specify on the `mmcrcluster` command or the `mmchcluster` command.

The default remote shell command is `ssh`. You can designate the use of a different remote shell command by specifying its fully qualified path name on the `mmcrcluster` command or the `mmchcluster` command. The remote shell command must adhere to the same syntax as `ssh`, but it can implement an alternate authentication mechanism.

Clusters that include both UNIX and Windows nodes must use `ssh` for the remote shell command. For more information, see [“Installing and configuring OpenSSH on Windows nodes”](#) on page 472.

Clusters that only include Windows nodes may use the `mmwinrsh` utility that comes with GPFS. The fully qualified path name is `/usr/lpp/mmfs/bin/mmwinrsh`. For more information about configuring Windows GPFS clusters, see **`mmwinrsh` command** in *IBM Spectrum Scale: Command and Programming Reference*.

By default, you can issue GPFS administration commands from any node in the cluster. Optionally, you can choose a subset of the nodes that are capable of running administrative commands. In either case, the nodes that you plan to use for administering GPFS must be able to run remote shell commands on any other node in the cluster as user `root` without the use of a password and without producing any extraneous messages. You can also use `sudo` wrappers for this purpose.

For additional information, see *Requirements for administering a GPFS file system* in *IBM Spectrum Scale: Administration Guide*.

Related concepts

GPFS node adapter interface names

An adapter interface name refers to the hostname or IP address that GPFS uses to communicate with a node. Specifically, the hostname or IP address identifies the communications adapter over which the GPFS daemons or GPFS administration commands communicate.

Creating an IBM Spectrum Scale cluster

When you create an IBM Spectrum Scale cluster, you can either create a small cluster of one or more nodes and later add nodes to it or you can create a cluster with all its nodes in one step.

IBM Spectrum Scale cluster configuration information

You can use the Cluster Configuration Repository (CCR) to maintain cluster configuration information.

Remote file copy command

The GPFS commands must maintain a number of configuration files across all nodes in the cluster. To achieve this, the GPFS commands use the remote file copy command that you specify on the `mmcrcluster` command or the `mmchcluster` command.

Cluster name

Provide a name for the cluster by issuing the `-C` option on the `mmcrcluster` command.

User ID domain for the cluster

This option is the user ID domain for a cluster when accessing a file system remotely.

Cluster configuration file

GPFS provides default configuration values, so a cluster configuration file is not required to create a cluster.

Related tasks

Starting GPFS automatically

You can specify whether to start the GPFS daemon automatically on a node when it is started.

Remote file copy command

The GPFS commands must maintain a number of configuration files across all nodes in the cluster. To achieve this, the GPFS commands use the remote file copy command that you specify on the `mmcrcluster` command or the `mmchcluster` command.

The default remote file copy program is **scp**. You can designate the use of a different remote file copy command by specifying its fully qualified path name on the **mmcrcluster** command or the **mmchcluster** command. The remote file copy command must adhere to the same syntax as **scp**, but it can implement an alternate authentication mechanism.

Clusters that include both UNIX and Windows nodes must use **scp** for the remote copy command. For more information, see [“Installing and configuring OpenSSH on Windows nodes”](#) on page 472.

The file copy command **rcp** cannot be used in a cluster that contains Windows Server nodes. However, clusters that include only Windows nodes may use the **mmwinrcp** utility that comes with GPFS. The fully qualified path name is `/usr/lpp/mmfs/bin/mmwinrcp`. For more information about configuring Windows GPFS clusters, see *mmwinservctl* command in *IBM Spectrum Scale: Command and Programming Reference*.

The nodes that you plan to use for administering GPFS must be able to copy files using the remote file copy command to and from any other node in the cluster without the use of a password and without producing any extraneous messages.

For additional information, see *Requirements for administering a GPFS file system* in *IBM Spectrum Scale: Administration Guide*.

Related concepts

[GPFS node adapter interface names](#)

An adapter interface name refers to the hostname or IP address that GPFS uses to communicate with a node. Specifically, the hostname or IP address identifies the communications adapter over which the GPFS daemons or GPFS administration commands communicate.

[Creating an IBM Spectrum Scale cluster](#)

When you create an IBM Spectrum Scale cluster, you can either create a small cluster of one or more nodes and later add nodes to it or you can create a cluster with all its nodes in one step.

[IBM Spectrum Scale cluster configuration information](#)

You can use the Cluster Configuration Repository (CCR) to maintain cluster configuration information.

[Remote shell command](#)

GPFS commands need to be able to communicate across all nodes in the cluster. To achieve this, the GPFS commands use the remote shell command that you specify on the `mmcrcluster` command or the `mmchcluster` command.

[Cluster name](#)

Provide a name for the cluster by issuing the `-C` option on the `mmcrcluster` command.

[User ID domain for the cluster](#)

This option is the user ID domain for a cluster when accessing a file system remotely.

[Cluster configuration file](#)

GPFS provides default configuration values, so a cluster configuration file is not required to create a cluster.

Related tasks

[Starting GPFS automatically](#)

You can specify whether to start the GPFS daemon automatically on a node when it is started.

Cluster name

Provide a name for the cluster by issuing the `-C` option on the `mmcrcluster` command.

If the user-provided name contains dots, it is assumed to be a fully qualified domain name. Otherwise, to make the cluster name unique in a multiple cluster environment, GPFS appends the domain name of the first node in the list primary cluster configuration server. If the `-C` option is not specified, the cluster name defaults to the hostname of the primary cluster configuration server. The name of the cluster may be changed at a later time by issuing the `-C` option on the `mmchcluster` command.

The cluster name is applicable when GPFS file systems are mounted by nodes belonging to other GPFS clusters. See the `mmauth` and the `mmremoteccluster` commands.

Related concepts

GPFS node adapter interface names

An adapter interface name refers to the hostname or IP address that GPFS uses to communicate with a node. Specifically, the hostname or IP address identifies the communications adapter over which the GPFS daemons or GPFS administration commands communicate.

Creating an IBM Spectrum Scale cluster

When you create an IBM Spectrum Scale cluster, you can either create a small cluster of one or more nodes and later add nodes to it or you can create a cluster with all its nodes in one step.

IBM Spectrum Scale cluster configuration information

You can use the Cluster Configuration Repository (CCR) to maintain cluster configuration information.

Remote shell command

GPFS commands need to be able to communicate across all nodes in the cluster. To achieve this, the GPFS commands use the remote shell command that you specify on the `mmcrcluster` command or the `mmchcluster` command.

Remote file copy command

The GPFS commands must maintain a number of configuration files across all nodes in the cluster. To achieve this, the GPFS commands use the remote file copy command that you specify on the `mmcrcluster` command or the `mmchcluster` command.

User ID domain for the cluster

This option is the user ID domain for a cluster when accessing a file system remotely.

Cluster configuration file

GPFS provides default configuration values, so a cluster configuration file is not required to create a cluster.

Related tasks

Starting GPFS automatically

You can specify whether to start the GPFS daemon automatically on a node when it is started.

User ID domain for the cluster

This option is the user ID domain for a cluster when accessing a file system remotely.

This option is further explained in *IBM Spectrum Scale: Administration Guide* and the IBM white paper *UID Mapping for GPFS in a Multi-cluster Environment* (https://www.ibm.com/docs/en/spectrum-scale?topic=STXKQY/uid_gpfs.pdf).

Related concepts

GPFS node adapter interface names

An adapter interface name refers to the hostname or IP address that GPFS uses to communicate with a node. Specifically, the hostname or IP address identifies the communications adapter over which the GPFS daemons or GPFS administration commands communicate.

Creating an IBM Spectrum Scale cluster

When you create an IBM Spectrum Scale cluster, you can either create a small cluster of one or more nodes and later add nodes to it or you can create a cluster with all its nodes in one step.

IBM Spectrum Scale cluster configuration information

You can use the Cluster Configuration Repository (CCR) to maintain cluster configuration information.

Remote shell command

GPFS commands need to be able to communicate across all nodes in the cluster. To achieve this, the GPFS commands use the remote shell command that you specify on the `mmcrcluster` command or the `mmchcluster` command.

Remote file copy command

The GPFS commands must maintain a number of configuration files across all nodes in the cluster. To achieve this, the GPFS commands use the remote file copy command that you specify on the `mmcrcluster` command or the `mmchcluster` command.

Cluster name

Provide a name for the cluster by issuing the `-C` option on the `mmcrcluster` command.

Cluster configuration file

GPFS provides default configuration values, so a cluster configuration file is not required to create a cluster.

Related tasks

Starting GPFS automatically

You can specify whether to start the GPFS daemon automatically on a node when it is started.

Starting GPFS automatically

You can specify whether to start the GPFS daemon automatically on a node when it is started.

Whether or not GPFS automatically starts is determined using the `autoload` parameter of the `mmchconfig` command. The default is *not* to automatically start GPFS on all nodes. You may change this by specifying `autoload=yes` using the `mmchconfig` command. This eliminates the need to start GPFS by issuing the `mmstartup` command when a node is booted.

The `autoload` parameter can be set the same or differently for each node in the cluster. For example, it may be useful to set `autoload=no` on a node that is undergoing maintenance since operating system upgrades and other software can often require multiple reboots to be completed.

Related concepts

GPFS node adapter interface names

An adapter interface name refers to the hostname or IP address that GPFS uses to communicate with a node. Specifically, the hostname or IP address identifies the communications adapter over which the GPFS daemons or GPFS administration commands communicate.

Creating an IBM Spectrum Scale cluster

When you create an IBM Spectrum Scale cluster, you can either create a small cluster of one or more nodes and later add nodes to it or you can create a cluster with all its nodes in one step.

IBM Spectrum Scale cluster configuration information

You can use the Cluster Configuration Repository (CCR) to maintain cluster configuration information.

Remote shell command

GPFS commands need to be able to communicate across all nodes in the cluster. To achieve this, the GPFS commands use the remote shell command that you specify on the `mmcrcluster` command or the `mmchcluster` command.

Remote file copy command

The GPFS commands must maintain a number of configuration files across all nodes in the cluster. To achieve this, the GPFS commands use the remote file copy command that you specify on the `mmcrcluster` command or the `mmchcluster` command.

Cluster name

Provide a name for the cluster by issuing the `-C` option on the `mmcrcluster` command.

User ID domain for the cluster

This option is the user ID domain for a cluster when accessing a file system remotely.

Cluster configuration file

GPFS provides default configuration values, so a cluster configuration file is not required to create a cluster.

Cluster configuration file

GPFS provides default configuration values, so a cluster configuration file is not required to create a cluster.

You can provide an optional cluster configuration file at the time of cluster creation. This optional file can be useful if you already know the correct parameter values based on previous testing or if you are restoring a cluster and have a backup copy of configuration values that apply to most systems. Typically, however, this option is not used at cluster creation time, and configuration parameters are modified after the cluster is created (using the `mmchconfig` command).

For more information about the cluster configuration file, see the description of the **`mmcrcluster -c ConfigFile`** option in *IBM Spectrum Scale: Command and Programming Reference*.

Related concepts

GPFS node adapter interface names

An adapter interface name refers to the hostname or IP address that GPFS uses to communicate with a node. Specifically, the hostname or IP address identifies the communications adapter over which the GPFS daemons or GPFS administration commands communicate.

Creating an IBM Spectrum Scale cluster

When you create an IBM Spectrum Scale cluster, you can either create a small cluster of one or more nodes and later add nodes to it or you can create a cluster with all its nodes in one step.

IBM Spectrum Scale cluster configuration information

You can use the Cluster Configuration Repository (CCR) to maintain cluster configuration information.

Remote shell command

GPFS commands need to be able to communicate across all nodes in the cluster. To achieve this, the GPFS commands use the remote shell command that you specify on the `mmcrcluster` command or the `mmchcluster` command.

Remote file copy command

The GPFS commands must maintain a number of configuration files across all nodes in the cluster. To achieve this, the GPFS commands use the remote file copy command that you specify on the `mmcrcluster` command or the `mmchcluster` command.

Cluster name

Provide a name for the cluster by issuing the `-C` option on the `mmcrcluster` command.

User ID domain for the cluster

This option is the user ID domain for a cluster when accessing a file system remotely.

Related tasks

Starting GPFS automatically

You can specify whether to start the GPFS daemon automatically on a node when it is started.

Disk considerations

Designing a proper storage infrastructure for your IBM Spectrum Scale file systems is key to achieving performance and reliability goals. When deciding what disk configuration to use, consider three key areas: infrastructure, performance, and disk access method.

Infrastructure

- Ensure that you have sufficient disks to meet the expected I/O load. In IBM Spectrum Scale terminology, a disk may be a physical disk or a RAID device.
- Ensure that you have sufficient connectivity (adapters and buses) between disks and network shared disk servers.
- Determine whether you are within IBM Spectrum Scale limits. Starting with GPFS 3.1, the structural limit on the maximum number of disks in a file system increased from 2048 to 4096; however, IBM Spectrum Scale still enforces the original limit of 2048. Should your environment require support for more than 2048 disks, contact the IBM Support Center to discuss increasing the enforced limit. (However, the number of disks in your system is often constrained by products other than IBM Spectrum Scale.)
- For a list of storage devices tested with IBM Spectrum Scale, see the [IBM Spectrum Scale FAQ in IBM Documentation](#).
- For Linux on Z, see the "Storage" topics "DASD device driver" and "SCSI-over-Fibre Channel device driver" in [Device Drivers, Features, and Commands](#) in the Linux on Z library overview.

Disk access method

- Decide how your disks will be connected. Supported types of disk connectivity include the following configurations:
 1. All disks SAN-attached to all nodes in all clusters that access the file system

In this configuration, every node sees the same disk simultaneously and has a corresponding disk device entry.
 2. Each disk connected to multiple NSD server nodes (up to eight servers), as specified on the server list

In this configuration, a single node with connectivity to a disk performs data shipping to all other nodes. This node is the first NSD server specified on the NSD server list. You can define additional NSD servers on the server list. Having multiple NSD servers guards against the loss of a single NSD server. When using multiple NSD servers, all NSD servers must have connectivity to the same disks. In this configuration, all nodes that are not NSD servers will receive their data over the local area network from the first NSD server on the server list. If the first NSD server fails, the next available NSD server on the list will control data distribution.
 3. A combination of SAN-attached and an NSD server configuration.

Configuration consideration:

- If the node has a physical attachment to the disk and that connection fails, the node switches to using a specified NSD server to perform I/O. For this reason, it is recommended that you define NSDs with multiple servers, even if all nodes have physical attachments to the disk.
- Configuring IBM Spectrum Scale disks without an NSD server stops the serving of data when the direct path to the disk is lost. This may be a preferable option for nodes requiring a higher speed data connection provided through a SAN as opposed to a lower speed network NSD server connection. Parallel jobs using MPI often have this characteristic.
- The `-o useNSDserver` file system mount option on the `mmount`, `mount`, `mmchfs`, and `mmremote fs` commands can be used to specify the disk discovery, and limit or eliminate switching from local access to NSD server access, or the other way around.

- Decide whether you will use storage pools to manage your disks.

Storage pools allow you to manage your file system's storage in groups. You can partition your storage based on such factors as performance, locality, and reliability. Files are assigned to a storage pool based on defined policies.

Policies provide for the following:

- Placing files in a specific storage pool when the files are created
- Migrating files from one storage pool to another
- File deletion based on file characteristics

For more information, see *Storage pools* in *IBM Spectrum Scale: Administration Guide*.

Network Shared Disk (NSD) creation considerations

You must prepare each physical disk you intend to use with GPFS by first defining it as a Network Shared Disk (NSD) using the `mmcrnsd` command.

On Windows, GPFS will only create NSDs from empty disk drives. `mmcrnsd` accepts Windows *Basic* disk or *Unknown/Not Initialized* disks. It always re-initializes these disks so that they become *Basic GPT Disks* with a single *GPFS partition*. NSD data is stored in GPFS partitions. This allows other operating system components to recognize that the disks are used. `mmdelnsd` deletes the partition tables created by `mmcrnsd`.

A new NSD format was introduced with GPFS 4.1. The new format is referred to as NSD v2, and the old format is referred to as NSD v1. The NSD v1 format is compatible with GPFS releases prior to 4.1. The latest GPFS release recognizes both NSD v1 and NSD v2 formatted disks.

The NSD v2 format provides the following benefits:

- On Linux, includes a partition table so that the disk is easily recognized as a GPFS device
- Adjusts data alignment to support disks with a 4 KB physical block size
- Adds backup copies of some key GPFS data structures
- Expands some reserved areas to allow for future growth

Administrators do not need to select one format or the other when managing NSDs. GPFS will always create and use the correct format based on the `minReleaseLevel` for the cluster and the file system version. When `minReleaseLevel` (as reported by `mmfsconfig`) is less than 4.1.0.0, `mmcrnsd` will only create NSD v1 formatted disks. When `minReleaseLevel` is at least 4.1.0.0, `mmcrnsd` will only create NSD v2 formatted disks. In this second case, however, the NSD format may change dynamically when the NSD is added to a file system so that the NSD is compatible with the file system version.

On Linux, NSD v2 formatted disks include a GUID Partition Table (GPT) with a single partition. The GPT allows other operating system utilities to recognize when a disk is owned by GPFS, which helps prevent inadvertent data corruption. After running `mmcrnsd`, Linux utilities like `parted` can show the partition table. When an NSD v2 formatted disk is added to a 3.5 or older file system, its format is changed to NSD v1 and the partition table is converted to an MBR (MS-DOS compatible) type.

Note: Leftover persistent reserve (PR) keys can cause problems such as reservation conflicts in multipath, which can in turn cause I/O failure. In such cases, it is necessary to clean up leftover PR keys on a fresh install. For a detailed procedure, see *Clearing a leftover Persistent Reserve reservation* in *IBM Spectrum Scale: Problem Determination Guide*.

The `mmcrnsd` command expects a stanza file as input. For details, see the following topics in *IBM Spectrum Scale: Command and Programming Reference* and *IBM Spectrum Scale: Administration Guide*:

- *Stanza files*
- **`mmchdisk`** command
- **`mmchnsd`** command
- **`mmcrfs`** command

- ***mmcrnsd*** command

Related concepts

IBM Spectrum Scale with data reduction storage devices

Data reduction is a feature in storage arrays that optimizes storage usage with techniques such as compression, deduplication, and thin provisioning to provide storage space to applications that exceeds the amount of physical space in the device.

NSD server considerations

If you plan to use NSD servers to remotely serve disk data to other nodes, as opposed to having disks SAN-attached to all nodes, you should consider the total computing and I/O load on these nodes.

File system descriptor quorum

A GPFS structure called the *file system descriptor* is initially written to every disk in the file system and is replicated on a subset of the disks as changes to the file system occur, such as the adding or deleting of disks.

Preparing direct access storage devices (DASD) for NSDs

Planning for GPUDirect Storage

IBM Spectrum Scale support for GPUDirect Storage (GDS) enables a direct path between GPU memory and storage. You need to ensure that certain conditions are met before you start installing the feature.

IBM Spectrum Scale with data reduction storage devices

Data reduction is a feature in storage arrays that optimizes storage usage with techniques such as compression, deduplication, and thin provisioning to provide storage space to applications that exceeds the amount of physical space in the device.

Related concepts

Network Shared Disk (NSD) creation considerations

You must prepare each physical disk you intend to use with GPFS by first defining it as a Network Shared Disk (NSD) using the `mmcrnsd` command.

NSD server considerations

If you plan to use NSD servers to remotely serve disk data to other nodes, as opposed to having disks SAN-attached to all nodes, you should consider the total computing and I/O load on these nodes.

File system descriptor quorum

A GPFS structure called the *file system descriptor* is initially written to every disk in the file system and is replicated on a subset of the disks as changes to the file system occur, such as the adding or deleting of disks.

Preparing direct access storage devices (DASD) for NSDs

Planning for GPUDirect Storage

IBM Spectrum Scale support for GPUDirect Storage (GDS) enables a direct path between GPU memory and storage. You need to ensure that certain conditions are met before you start installing the feature.

IBM Spectrum Scale with thin provisioned devices

IBM Spectrum Scale provides features to reduce the risks of thin provisioning and to optimize storage use in thin provisioned storage systems.

With thin provisioning, storage space can be optimized by allocating space on demand and reclaiming when the application using a given volume declares it is no longer using a given block of data. The storage system allocates storage to applications, including file systems, only when the storage is needed, from a pool of free storage. Thin provisioning addresses the problem of under-utilization of available storage system capacity. System administrators do not have to monitor and manage how much storage is required by each application. Instead, thin provisioning lets administrators provision large "thin" LUNs to a host based on estimated and anticipated usage without reserving physical space in advance. When the application writes data, the storage system allocates physical storage from the free pool on the storage system to the thin-provisioned LUNs.

Thin provisioning of storage for file systems can provide an excellent alternative in increasingly cost-conscious data centers by allowing storage hardware purchases to be delayed and capacity to be purchased just in time.

A significant concern with data reduction storage is the possibility that the device might run out of physical space on a thin provisioned volume. For a file system running on such a device, the expectation is that the volume has been fully allocated, but if the storage has been over-provisioned, it is possible that no physical space exists to accommodate a write request. The result is that the device returns an I/O error, an event that can cause the file system to unmount and become inaccessible. In the case of an IBM Spectrum Scale file system, re-mounting the file system requires running through log recovery, a process that itself requires writing into the file system, and that might again fail if the device has no physical disk space available.

To minimize the possibility of the device running out of physical space, the device needs to be vigilantly monitored; otherwise the benefits of thin provisioning might be offset by the possibility the device will run out of space and bring the file system offline.



Attention: Data reduction in IBM Spectrum Scale, including thin provisioned volumes, is supported only after a technical compatibility review by IBM. Ask your sales representative to contact IBM Spectrum Scale development about the RPQ or SCORE process.

Enhancement in IBM Spectrum Scale

Spectrum Scale provides several features to mitigate the risks of supporting thin provisioning and to optimize storage use on thin provisioned storage systems:

- Space reclamation: This operation ensures that disk blocks that are not used by the file system are returned to the storage device.
- Emergency disk space management: Space is reserved in the file system, in the form of metadata files with disk blocks filled with uncompressible data, that can be freed back to the storage on demand.

Table 23. Thin provisioning storage solutions in IBM Spectrum Scale		
Feature	Description	Benefits
Automatic recovery space. For more information see the description of the three types of recovery space in the subtopic “Deployment recommendation” on page 242.	Automatic recovery space is reserved by the file system, which enables IBM Spectrum Scale to recover the storage system if it runs low on physical space.	This feature reduces system administrator workload to estimate how much additional physical space is needed to prevent a file system from becoming inaccessible when a storage system runs low on physical space.
The recovery log flush operation is thin-space aware.	The recovery log flush operation ensures that enough reserved space is always available to run log recovery.	This feature ensures that a file system can be recovered after a storage system runs out of physical space.
Thin space reclamation is supported.	The mmreclaimspace command reclaims unused space in a file system and returns it to the storage system.	This feature improves storage space utilization and savings.
Thin space emergency recovery.	Free the previously reserved space to allow the storage system to acquire enough space for file system recovery.	This feature recovers file systems after a failure caused by an out-of-physical-space condition in the storage system.

Deployment recommendation

Three types of emergency recovery space are used:

- *Required recovery space:* You must reserve this space before you create an IBM Spectrum Scale file system. Create several volumes in the storage system and fill them with uncompressible data. In the event that the storage system runs out of physical space, the required recovery space enables IBM Spectrum Scale to recover the volumes in the storage system back to a read-write state. For more information see the subtopic [“During system configuration” on page 242](#) later in this help topic.
- *Optional recovery space:* It is a good practice to also create several additional recovery volumes and fill them with uncompressible data. If the storage system runs nearly out of physical space, the optional recovery space enables IBM Spectrum Scale to prevent the storage system from running completely out of space. For more information see the subtopic [“During system configuration” on page 242](#) later in this help topic.
- *Automatic recovery space:* IBM Spectrum Scale reserves this space for a file system the first time that the file system is mounted. This type of space enables IBM Spectrum Scale to recover the file system if the storage system runs out of space. However, this type of space can be released to IBM Spectrum Scale only while the device is operational and in read-write mode. It cannot be released if the file system has become either read-only or offline because the storage system has run completely out of space. For more information see the subtopic [“Emergency recovery” on page 243](#) later in this subtopic.

The following subtopics describe the steps that you should take for thin provisioning support during system configuration, during ongoing operations (production), and during emergency recovery after the storage system has run nearly or completely out of physical space.

During system configuration

During system configuration you must reserve a *required recovery space* and you can also create an *optional recovery space*. For the meaning of these terms see the previous subtopic.

To create the required recovery space, create several emergency recovery volumes in the storage system and fill them with uncompressible data, such as fully random data that cannot be subject to deduplication. The size of this space varies depending on the type and brand of the storage system. For more information about the amount of reserved space that is needed, contact the vendor of the storage system. If you are using an IBM FlashSystem A9000, A9000R or any IBM Spectrum Virtualize systems using Flash Core Modules (FCMs) and run out of space, contact IBM Spectrum Virtualize support team for assistance.

If the storage system runs out of physical space, IBM Spectrum Scale uses this required recovery space to bring the storage system back online. For more information see the subtopic [“Emergency recovery” on page 243](#) later in this help topic.

After you create the required recovery space, you must create LUNs in the storage system, export them, and specify each LUN in an nsd stanza in the IBM Spectrum Scale stanza file. See the following example:

```
%nsd:
  nsd=gpfs1nsd
  usage=metadataOnly
  pool=system
  thinDiskType=scsi
```

Figure 32. Specifying a stanza for a thin-provisioned disk

The line `thinDiskType=scsi` is required and indicates that the device is a thin provisioned disk.

Note: The device must support the SCSI WRITE SAME command.

For more information, see the topics *mmcrnsd command* and *mmcrfs command* in the *IBM Spectrum Scale: Command and Programming Reference*.

To create the optional recovery space, create several volumes in the storage system and fill them with uncompressible data. A suggested size for the emergency recovery space is 20 GiB times the sum of all

the local and remote nodes that can mount each of the file systems that is located on the storage system. In other words:

```
20 GiB x ((the number of local and remote nodes that can mount FS1) +  
          (the number of local nodes that can mount FS2) +  
          .... +  
          (the number of local and remote nodes that can mount FSN))
```

If the storage system runs nearly out of physical space, IBM Spectrum Scale can use this optional recovery space to prevent the storage system from running completely out of physical space. For more information see the subtopic [“Emergency recovery” on page 243](#) later in this help topic.

During production

During production you must carefully monitor the amount of physical space that is available in the storage system. The vendor of the storage system can probably provide utility programs for monitoring available physical space. If you determine that the storage system is running nearly out of physical space, follow the steps below. (If the storage system runs completely out of physical space, follow the instructions in the [“Emergency recovery” on page 243](#) subtopic later in this help topic.)

1. Quiesce all applications that are writing data into the storage system.
2. Delete unused volumes from the storage system.
3. Delete unused files and directories in the IBM Spectrum Scale file systems.
4. If you created optional recovery space during system configuration, free it. With this freed space IBM Spectrum Scale can recover the storage system before it runs completely out of space.
5. Issue the **mmreclaimspace** command to reclaim physical space in the storage system that is not in use but is not marked as available:

```
mmreclaimspace Device --reclaim-threshold 0
```

For more information, see the topic *mmreclaimspace command* in the *IBM Spectrum Scale: Command and Programming Reference*.

Note: Because the space-reclaiming operation consumes a great deal of CPU, memory, network, and I/O system resources, it is a good idea to run the **mmreclaimspace** command at a time when the system is not heavily loaded.

6. Expand the free physical space in the storage system by adding new disks or deleting existing volumes.

If the storage system now has enough available physical space, you can resume all applications. If you freed the optional recovery space in Step 4, it is a good practice to recreate it as soon as possible.

Emergency recovery

If the storage system runs completely out of physical space, the volumes that use the storage system are likely to go offline or become read-only. To recover the system storage and to recover the IBM Spectrum Scale file systems, follow the procedure that is outlined below. This procedure has three main parts:

Part 1: Bring the file system to a state in which data can be read although not written. Follow these steps:

1. Stop all applications that are writing into the storage system.
2. Disable the operating system automount feature by issuing the appropriate system command:
 - On Linux, issue the following command:

```
systemctl disable autofs
```

- In AIX, issue the following command:

```
stopsrc -s automountd
```

3. Issue the following command to unmount all IBM Spectrum Scale file systems:

```
/usr/lpp/mmfs/bin/mmumount all -a
```

4. Release the required recovery space that you reserved during system configuration. All volumes should be operational and in read-write mode.
5. Issue the following command to correct the states of the NSD disks. This action allows the mount operation in Step 7 to succeed across the cluster:

```
/usr/lpp/mmfs/bin/mmnsddiscover -a -N all
```

6. If you created optional recovery space during system configuration, free it.
7. Remount the file system in read-only mode. If the mount succeeds, skip the next step and go on to Part 2. IBM Spectrum Scale reserves the automatic recovery space as the file system is mounted for the first time.
8. If the mount operation in the preceding step fails, follow these steps to mount the file system:
 - a. Issue the following command to mount the file system in restricted mode:

```
/usr/lpp/mmfs/bin/mmmount Device -o rs
```

- b. Issue the following command to release the automatic recovery space that IBM Spectrum Scale reserved when it created the file system:

```
/usr/lpp/mmfs/bin/mmreclaimspace Device --emergency-reclaim
```

Note: The **--emergency-reclaim** option is valid only if the thin-provisioned storage is read-writable and the file system is mounted in restrict mode. For more information, see the topic *mmreclaimspace command* in the *IBM Spectrum Scale: Command and Programming Reference*.

- c. Remount the file system in read-only mode as in Step 7. The file system should be mounted successfully.

Part 2: Expand the free physical space of the storage system. You can add disks, delete data, or move data to other file systems.

Part 3: Bring the file system to a state in which data can be both read and written. Follow these steps:

1. Reserve required recovery space in the physical storage system as you did during system configuration.
2. If you freed optional recovery space in Step 6 of Part 1, recreate it as you did during system configuration.
3. Issue the following command to mount the file system in read-write mode:

```
/usr/lpp/mmfs/bin/mmmount Device
```

If you ran the **mmreclaimspace** command in Step 8(b) of Part 1, IBM Spectrum Scale reserves automatic recovery space during this mount operation.

4. Enable the operating system automount feature by issuing the appropriate system command:

- On Linux issue the following command:

```
systemctl enable autofs
```

- In AIX issue the following command:

```
startsrc -s automountd
```

If Step 3 completed without any error, the file system is writable again. If an error occurred, the free physical space that you expanded in Part 2 is not enough. Repeat the emergency recovery process, beginning with Part 1, Step 1.

IBM Spectrum Scale with TRIM-supporting NVMe SSDs

IBM Spectrum Scale provides features that address the performance issues of flash memory.

File systems that create and remove files and directories often reuse storage blocks by overwriting the same storage blocks with new data contents. However, an NVMe or a general solid-state drive (SSD) device cannot overwrite a page of flash storage without first erasing the entire block of storage in which the page is located. This behavior creates a performance issue for I/O writes to previously used blocks of data when compared with I/O writes to unused or erased blocks. Also, write amplification occurs when the drive must save a copy of existing pages that are unaffected by the I/O write operation, then erase the entire block, and then restore the unaffected pages to the block after the erase. To improve performance, the file system can issue a TRIM operation to an NVMe SSD notifying it about which blocks of data are no longer in use and can therefore be erased. The NVMe SSD erases the unused blocks before the blocks are required for the next reuse, which improves the performance of future I/O writes to the device. The TRIM operation also reduces fragmentation, because unused blocks are erased.

For more information about write amplification, garbage collection, and performance issues, see the vendor specifications for the flash memory devices that you are using.

Deployment recommendation

In the IBM Spectrum Scale stanza file, specify the `thinDiskType=nvme` line in each `nsd` stanza that describes an NVMe device:

```
%nsd:
  nsd=gpfs1nsd
  usage=metadataOnly
  pool=system
  thinDiskType=nvme
```

This line indicates to IBM Spectrum Scale that the disk is trim-capable.

Enhancement in IBM Spectrum Scale

IBM Spectrum Scale provides the **mmreclaimspace** command, which calls the TRIM command to notify the specified SSD device as to which blocks are no longer in use and can be erased. For more information, see the topic *mmreclaimspace* in the *IBM Spectrum Scale: Command and Programming Reference*.

Note: Because the space-reclaiming operation consumes a great deal of CPU, memory, network, and I/O system resources, it is a good idea to run the **mmreclaimspace** command at a time when the system is not heavily loaded.

Support matrix

The following conditions must be met for thin provisioning support to be activated:

- The file system must be upgraded to file system format version 5.0.4 or later.
- The stanza file must include the following line to add thin disks or NVMe SSDs into the file system. See the example `nsd` stanza in the preceding subtopic, *Deployment recommendation*:

```
thinDiskType={scsi | nvme}
```

- Both thin provisioned disks and NVMe SSDs must be connected to nodes that are running the Linux operating system.

Table 24. Supported operating systems and device connections

Operating system	A node can act as an NSD server or as a node with a disk directly attached	A node can act as a client
Linux (RHEL 7.3 and later SLES 11 and later, Ubuntu 16 and later)	Yes	Yes
Linux on Z	No	Yes
AIX	No	Yes
Windows	No	Yes

Table 25. Supported devices

Supported devices
<p>The following storage devices are supported:</p> <ul style="list-style-type: none"> • IBM A9000/R • IBM v7K • Intel NVMe SSDs
<p>The following devices are believed to operate correctly with IBM Spectrum Scale but support has not been fully verified:</p> <ul style="list-style-type: none"> • IBM FS9100 • General SSDs

NSD server considerations

If you plan to use NSD servers to remotely serve disk data to other nodes, as opposed to having disks SAN-attached to all nodes, you should consider the total computing and I/O load on these nodes.

- Will your Network Shared Disk servers be dedicated servers or will you also be using them to run applications? If you will have non-dedicated servers, consider running less time-critical applications on these nodes. If you run time-critical applications on a Network Shared Disk server, servicing disk requests from other nodes might conflict with the demands of these applications.
- The special functions of the file system manager consume extra processing time. If possible, avoid using a Network Shared Disk server as the file system manager. The Network Shared Disk server consumes both memory and processor cycles that could impact the operation of the file system manager. See [“The file system manager” on page 9](#).
- The actual processing capability required for Network Shared Disk service is a function of the application I/O access patterns, the type of node, the type of disk, and the disk connection. You can later run `iostat` on the server to determine how much of a load your access pattern will place on a Network Shared Disk server.
- Providing sufficient disks and adapters on the system to yield the required I/O bandwidth. Dedicated Network Shared Disk servers should have sufficient disks and adapters to drive the I/O load you expect them to handle.
- Knowing approximately how much storage capacity you will need for your data.

You should consider what you want as the default behavior for switching between local access and NSD server access in the event of a failure. To set this configuration, use the `-o useNSDserver` file system mount option of the `mmmount`, `mount`, `mmchfs`, and `mmremotefs` commands to:

- Specify the disk discovery behavior

- Limit or eliminate switching from either:
 - Local access to NSD server access
 - NSD server access to local access

You should consider specifying how long to wait for an NSD server to come online before allowing a file system mount to fail because the server is not available. The `mmchconfig` command has these options:

nsdServerWaitTimeForMount

When a node is trying to mount a file system whose disks depend on NSD servers, this option specifies the number of seconds to wait for those servers to come up. If a server recovery is taking place, the wait time you are specifying with this option starts after recovery completes.

Note: The decision to wait for servers is controlled by the `nsdServerWaitTimeWindowOnMount` option.

nsdServerWaitTimeWindowOnMount

Specifies a window of time (in seconds) during which a mount can wait for NSD servers as described for the `nsdServerWaitTimeForMount` option. The window begins when quorum is established (at cluster startup or subsequently), or at the last known failure times of the NSD servers required to perform the mount.

Notes:

1. When a node rejoins a cluster, it resets all the failure times it knew about within that cluster.
2. Because a node that rejoins a cluster resets its failure times within that cluster, the NSD server failure times are also reset.
3. When a node attempts to mount a file system, GPFS checks the cluster formation criteria first. If that check falls outside the window, it will then check for NSD server fail times being in the window.

Related concepts

[Network Shared Disk \(NSD\) creation considerations](#)

You must prepare each physical disk you intend to use with GPFS by first defining it as a Network Shared Disk (NSD) using the `mmcrnsd` command.

[IBM Spectrum Scale with data reduction storage devices](#)

Data reduction is a feature in storage arrays that optimizes storage usage with techniques such as compression, deduplication, and thin provisioning to provide storage space to applications that exceeds the amount of physical space in the device.

[File system descriptor quorum](#)

A GPFS structure called the *file system descriptor* is initially written to every disk in the file system and is replicated on a subset of the disks as changes to the file system occur, such as the adding or deleting of disks.

[Preparing direct access storage devices \(DASD\) for NSDs](#)

[Planning for GPUDirect Storage](#)

IBM Spectrum Scale support for GPUDirect Storage (GDS) enables a direct path between GPU memory and storage. You need to ensure that certain conditions are met before you start installing the feature.

File system descriptor quorum

A GPFS structure called the *file system descriptor* is initially written to every disk in the file system and is replicated on a subset of the disks as changes to the file system occur, such as the adding or deleting of disks.

Based on the number of failure groups and disks, GPFS creates one to five replicas of the descriptor:

- If there are at least five different failure groups, five replicas are created.
- If there are at least three different disks, three replicas are created.
- If there are only one or two disks, a replica is created on each disk.

Once it decides how many replicas to create, GPFS picks disks to hold the replicas, so that all replicas are in different failure groups, if possible, to reduce the risk of multiple failures. In picking replica locations, the current state of the disks is taken into account. Stopped or suspended disks are avoided. Similarly, when a failed disk is brought back online, GPFS might rebalance the file system descriptors in order to assure reliability across the failure groups. The disks used to hold the file system descriptor replicas can be seen by running the `mmfslsdisk fsname -L` command and looking for the string `desc` in the `remarks` column.

GPFS requires that a majority of the replicas on the subset of disks remain available to sustain file system operations:

- If there are at least five different replicas, GPFS can tolerate a loss of two of the five replicas.
- If there are at least three replicas, GPFS can tolerate a loss of one of the three replicas.
- If there are fewer than three replicas, a loss of one replica might make the descriptor inaccessible.

The loss of all disks in a disk failure group might cause a majority of file systems descriptors to become unavailable and inhibit further file system operations. For example, if your file system is backed up by three or more disks that are assigned to two separate disk failure groups, one of the failure groups will be assigned two of the file system descriptor replicas, while the other failure group will be assigned only one replica. If all of the disks in the disk failure group that contains the two replicas were to become unavailable, the file system would also become unavailable. To avoid this particular scenario, you might want to introduce a third disk failure group consisting of a single disk that is designated as a `descOnly` disk. This disk would exist solely to contain a replica of the file system descriptor (that is, it would not contain any file system metadata or data). This disk should be at least 128MB in size.

For more information, see [“Network Shared Disk \(NSD\) creation considerations”](#) on page 239 and *Establishing disaster recovery for your GPFS cluster* in *IBM Spectrum Scale: Administration Guide*.

Related concepts

[Network Shared Disk \(NSD\) creation considerations](#)

You must prepare each physical disk you intend to use with GPFS by first defining it as a Network Shared Disk (NSD) using the `mmcrnsd` command.

[IBM Spectrum Scale with data reduction storage devices](#)

Data reduction is a feature in storage arrays that optimizes storage usage with techniques such as compression, deduplication, and thin provisioning to provide storage space to applications that exceeds the amount of physical space in the device.

[NSD server considerations](#)

If you plan to use NSD servers to remotely serve disk data to other nodes, as opposed to having disks SAN-attached to all nodes, you should consider the total computing and I/O load on these nodes.

[Preparing direct access storage devices \(DASD\) for NSDs](#)

[Planning for GPUDirect Storage](#)

IBM Spectrum Scale support for GPUDirect Storage (GDS) enables a direct path between GPU memory and storage. You need to ensure that certain conditions are met before you start installing the feature.

Preparing direct access storage devices (DASD) for NSDs

When preparing direct access storage devices (DASD) for NSDs, see the table "Disk hardware tested with GPFS for Linux on Z" in the [IBM Spectrum Scale FAQ in IBM Documentation](#).

Related concepts

[Network Shared Disk \(NSD\) creation considerations](#)

You must prepare each physical disk you intend to use with GPFS by first defining it as a Network Shared Disk (NSD) using the `mmcrnsd` command.

[IBM Spectrum Scale with data reduction storage devices](#)

Data reduction is a feature in storage arrays that optimizes storage usage with techniques such as compression, deduplication, and thin provisioning to provide storage space to applications that exceeds the amount of physical space in the device.

NSD server considerations

If you plan to use NSD servers to remotely serve disk data to other nodes, as opposed to having disks SAN-attached to all nodes, you should consider the total computing and I/O load on these nodes.

File system descriptor quorum

A GPFS structure called the *file system descriptor* is initially written to every disk in the file system and is replicated on a subset of the disks as changes to the file system occur, such as the adding or deleting of disks.

Planning for GPUDirect Storage

IBM Spectrum Scale support for GPUDirect Storage (GDS) enables a direct path between GPU memory and storage. You need to ensure that certain conditions are met before you start installing the feature.

Preparing your environment for use of extended count key data (ECKD) devices

If your GPFS cluster includes Linux on Z instances, do not use virtual reserve/release.

Instead, follow the process that is described in [Sharing DASD without Using Virtual Reserve/Release \(https://www.ibm.com/docs/en/zvm/7.1?topic=sharing-dasd-without-using-virtual-reserverelease\)](https://www.ibm.com/docs/en/zvm/7.1?topic=sharing-dasd-without-using-virtual-reserverelease). Data integrity is handled by GPFS itself.

Preparing an ECKD device for GPFS

To prepare an ECKD device for GPFS, complete these steps on a single node:

1. Ensure that the ECKD device is online. To set it online, issue the following command:

```
chccwdev -e device_bus_id
```

Where *device_bus_id* identifies the device to be configured. *device_bus_id* is a device number with a leading 0.*n*, where *n* is the subchannel set ID. For example:

```
chccwdev -e 0.0.3352
```

2. Low-level format the ECKD using one of the following commands.

Note: GPFS supports ECKD disks in either compatible disk layout (CDL) format or Linux disk layout (LDL) format. The DASD must be formatted with a block size of 4096.

- To specify CDL format, issue the following command:

```
dasdfmt -d cdl device
```

There is no need to specify a block size value, as the default value is 4096.

- To specify LDL format, issue the following command:

```
dasdfmt -d ldl device
```

There is no need to specify a block size value, as the default value is 4096.

In both of these commands, *device* is the node of the device. For example:

```
dasdfmt -d cdl /dev/dasda
```

3. This step is for *CDL disks only*. It is an *optional* step because partitioning is optional for CDL disks.

If you want to partition the ECKD and create a single partition that spans the entire device, use the following command:

```
fdasd -a device
```

Note: This step is not required for LDL disks because the `dasdfmt -d 1d1` command issued in the previous step automatically creates a single Linux partition on the disk.

4. If the ECKD device is shared with other cluster nodes, then it is a good idea to set the ECKD device offline and then back online on each node where the ECKD is shared. This procedure ensures that the updated partition information is spread among all shared nodes.

For more information about all of these commands, see the following:

- "Commands for Linux on Z" topic in [Device Drivers, Features, and Commands \(https://www.ibm.com/docs/en/linux-on-systems?topic=configuration-device-drivers-features-commands\)](https://www.ibm.com/docs/en/linux-on-systems?topic=configuration-device-drivers-features-commands) in the Linux on Z library overview.
- "Getting started with Elastic Storage for Linux on Z based on GPFS technology" white paper, available on the Welcome Page for [IBM Spectrum Scale in IBM Documentation](#).

Repeat these steps for each ECKD to be used with GPFS.

After preparing the environment, set the ECKD devices online on the other nodes.

Note: After partitioning and formatting an ECKD device, the partition information is not distributed automatically to other cluster nodes. If the ECKD device is shared with different cluster nodes when they are online. It is recommended to set the ECKD device offline and then back online. This procedure updates the partition information for the specific ECKD device on the cluster node.

Always ensure that the ECKD devices are online before starting GPFS. To automatically set ECKD devices online at system start, see the documentation for your Linux distribution.

Planning for GPUDirect Storage

IBM Spectrum Scale support for GPUDirect Storage (GDS) enables a direct path between GPU memory and storage. You need to ensure that certain conditions are met before you start installing the feature.

For more information on GDS, see [“GPUDirect Storage support for IBM Spectrum Scale” on page 26](#).

The following list provides the prerequisite for installing and using GDS:

- **IBM Spectrum Scale version:** Requires at IBM Spectrum Scale 5.1.2 (IB) or 5.1.3 (RoCE). For more details on supported versions, see [IBM Spectrum Scale FAQ in IBM Documentation](#). If you need to configure GDS for a remote file system, IBM Spectrum Scale 5.1.2 or later is required on all NSD servers of the remote cluster.
- **Data path:** GDS requires the NSD path for storage access. The storage must not be locally attached to the GDS clients.
- **Storage:** ESS and non-ESS storage servers are supported. The storage servers can be in the same or different cluster as the GDS clients.
- **InfiniBand fabric:** GDS requires Mellanox RDMA over InfiniBand between GDS clients and storage servers.
- **Adapters:** Mellanox ConnectX-5 or newer adapters are required to use GDS with RoCE.

Supported hardware

The following list provides the hardware requirements:

- **GDS clients:** Intel x86 with a GPU model that supports GDS. For more details, see [NVIDIA GDS documentation](#).
- **Network:** EDR or HDR Infiniband, Ethernet (RoCE).
- **InfiniBand adapter:** CX5 or CX6. (Mellanox CX4 is OK for IB only if the CX4 firmware is 12.27.4000 or higher.)

Supported software for GDS clients

For information about the supported versions of the required software for GDS clients, see [Components required for GDS in IBM Spectrum Scale FAQ in IBM Documentation](#).

Supported software for GDS servers

The following list provides the software requirements:

- Any operating system that supports IBM Spectrum Scale can be used.
- MOFED: Use the Mellanox OFED stack for the OS. For details about driver versions, see [Components required for GDS in IBM Spectrum Scale FAQ in IBM Documentation](#).

Related concepts

[Network Shared Disk \(NSD\) creation considerations](#)

You must prepare each physical disk you intend to use with GPFS by first defining it as a Network Shared Disk (NSD) using the `mmcrnsd` command.

[IBM Spectrum Scale with data reduction storage devices](#)

Data reduction is a feature in storage arrays that optimizes storage usage with techniques such as compression, deduplication, and thin provisioning to provide storage space to applications that exceeds the amount of physical space in the device.

[NSD server considerations](#)

If you plan to use NSD servers to remotely serve disk data to other nodes, as opposed to having disks SAN-attached to all nodes, you should consider the total computing and I/O load on these nodes.

[File system descriptor quorum](#)

A GPFS structure called the *file system descriptor* is initially written to every disk in the file system and is replicated on a subset of the disks as changes to the file system occur, such as the adding or deleting of disks.

[Preparing direct access storage devices \(DASD\) for NSDs](#)

[“GPUDirect Storage support for IBM Spectrum Scale” on page 26](#)

IBM Spectrum Scale's support for NVIDIA's GPUDirect Storage (GDS) enables a direct path between GPU memory and storage. This solution addresses the need for higher throughput and lower latencies. File system storage is directly connected to the GPU buffers to reduce latency and load on CPU. For IBM Spectrum Scale, this means that data can be read directly from an NSD server's pagepool and it is sent to the GPU buffer of the IBM Spectrum Scale clients by using RDMA. IBM Spectrum Scale with GDS requires an InfiniBand or RoCE fabric. In IBM Spectrum Scale, the `mmdiag` command is enhanced to print diagnostic information for GPUDirect Storage.

Related tasks

[“Installing GPUDirect Storage for IBM Spectrum Scale” on page 481](#)

IBM Spectrum Scale support for GPUDirect Storage (GDS) enables a direct path between GPU memory and storage.

[“Upgrading GPUDirect Storage” on page 517](#)

You need to upgrade your IBM Spectrum Scale cluster to 5.1.2 or later to start using the GPUDirect Storage (GDS).

File system creation considerations

File system creation involves anticipating usage within the file system and considering your hardware configurations. Before creating a file system, consider how much data will be stored and how great is the demand for the files in the system.

Each of these factors can help you to determine how much disk resource to devote to the file system, which block size to choose, where to store data and metadata, and how many replicas to maintain. For the latest supported file system size, see the [IBM Spectrum Scale FAQ in IBM Documentation](#).

Your GPFS file system is created by issuing the `mmcrfs` command. Table 26 on page 252 details the file system creation options that are specified on the `mmcrfs` command, which options can be changed later with the `mmchfs` command, and what the default values are.

To move an existing file system into a new GPFS cluster, see *Exporting file system definitions between clusters* in *IBM Spectrum Scale: Administration Guide*.

Table 26. File system creation options			
Options	mmcrfs	mmchfs	Default value
<code>--auto-inode-limit</code> automatically increases the maximum number of inodes per inode space in the file system See mmcrfs command .	X	X	<code>--noauto-inode-limit</code>
Device name of the file system. See “Device name of the file system” on page 257.	X	X	none
DiskDesc for each disk in your file system. Note: The use of disk descriptors is discouraged. See “Disks for your file system” on page 257.	X	Issue the <code>mmadddisk</code> or <code>mmdeledisk</code> command to add or delete disks from the file system.	none
<code>-F StanzaFile</code> specifies a file that contains a list of NSD stanzas and pool stanzas. For more information, see <i>Stanza files</i> in <i>IBM Spectrum Scale: Administration Guide</i> .	X	Issue the <code>mmadddisk</code> or <code>mmdeledisk</code> command to add or delete disks as indicated in the stanza file.	none
<code>-A {yes no automount}</code> to determine when to mount the file system. See “Deciding how the file system is mounted” on page 257.	X	X	yes

Table 26. File system creation options (continued)

Options	mmcrfs	mmchfs	Default value
<p>-B BlockSize</p> <ul style="list-style-type: none"> By default, all the data blocks and metadata blocks in a file system are set to the same block size with the same subblock size. Metadata blocks can be set to a different block size with the <code>--metadata-block-size</code> parameter, and this setting can change the size and number of subblocks in data blocks. <p>See “Block size” on page 258 and <i>mmcrfs</i> command in <i>IBM Spectrum Scale: Command and Programming Reference</i>.</p>	X	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.	4 MiB
<p>-D {posix nfs4} semantics for a deny-write open lock</p> <p>See “NFS V4 deny-write open lock” on page 257.</p>	X	X	nfs4
<p>-E {yes no} to report exact mtime values.</p> <p>See “mtime values” on page 261.</p>	X	X	yes
<p>--flush-on-close specifies the automatic flushing of disk buffers when closing files that were opened for write operations on the device.</p> <p>See <i>mmcrfs</i> command</p>	X	X	--noflush-on-close
<p>-i InodeSize to set the size of inodes: 512, 1024, or 4096 bytes.</p>	X	This value cannot be changed.	4096
<p>-j {cluster scatter} to determine the block allocation map type.</p> <p>See “Block allocation map” on page 261.</p>	X	NA	See “Block allocation map” on page 261.
<p>-k {posix nfs4 all} to determine the authorization types supported by the file system.</p> <p>See “File system authorization” on page 262.</p>	X	X	all
<p>-K {no whenpossible always} to enforce strict replication.</p> <p>See “Type of replication” on page 262.</p>	X	X	whenpossible

Table 26. File system creation options (continued)

Options	mmcrfs	mmchfs	Default value
-L <i>LogFileSize</i> to specify the size of the internal log files. See “GPFS recovery logs” on page 13.	X	X	See the topic <i>mmcrfs</i> command in the <i>IBM Spectrum Scale: Command and Programming Reference</i> .
-m <i>DefaultMetadataReplicas</i> See “File system replication parameters” on page 263.	X	X	1
-M <i>MaxMetadataReplicas</i> See “File system replication parameters” on page 263.	X	This value cannot be changed.	2
-n <i>NumNodes</i> that will mount the file system. See “Number of nodes mounting the file system” on page 264.	X	X	32
--nfs4-owner-write-acl specifies whether object owners are given implicit NFSv4 WRITE_ACL permission. See <i>mmcrfs</i> command	X	X	yes
-o <i>MountOptions</i> to be passed to the mount command. See “Assign mount command options” on page 265.	NA	X	none
-Q {yes no} to activate quota. See “Enabling quotas” on page 265.	X	X	no
-r <i>DefaultDataReplicas</i> See “File system replication parameters” on page 263.	X	X	1
-R <i>MaxDataReplicas</i> See “File system replication parameters” on page 263.	X	This value cannot be changed.	2
-S {yes no relatime} to control how the atime value is updated. See “atime values” on page 260.	X	X	The default value is <i>relatime</i> if <i>minReleaseLevel</i> is 5.0.0 or later when the file system is created. Otherwise the default value is <i>no</i> .

Table 26. File system creation options (continued)

Options	mmcrfs	mmchfs	Default value
-t <i>DriveLetter</i> See “Windows drive letter” on page 265.	X	X	none
-T <i>Mountpoint</i> See “Mountpoint directory” on page 265.	X	X	/gpfs/DeviceName
-V {full compat} to change the file system format to the latest level. See “Changing the file system format to the latest level” on page 266	NA	X	none
-v {yes no} to verify disk usage. See “Verifying disk usage” on page 266.	X	NA	yes
-W <i>NewDeviceName</i> to assign a new device name to the file system.	NA	X	none
-z {yes no} to enable DMAPI See “Enabling DMAPI” on page 266.	X	X	no
--filesetdf to specify (when quotas are enforced for a fileset) whether the df command will report numbers based on the quotas for the fileset and not for the total file system.	X	X	--nofilesetdf
--inode-limit <i>MaxNumInodes</i> [: <i>NumInodesToPreallocate</i>] to determine the maximum number of files in the file system. See “Specifying the maximum number of files that can be created” on page 267.	X	X	file system size/1 MiB
--log-replicas <i>LogReplicas</i> to specify the number of recovery log replicas.	X	X	none

Table 26. File system creation options (continued)

Options	mmcrfs	mmchfs	Default value
--metadata-block-size <i>MetadataBlockSize</i> Specifies a metadata block size that is different from the data block size. This parameter requires a separate pool for metadata. See “Block size” on page 258 and <i>mmcrfs</i> command in <i>IBM Spectrum Scale: Command and Programming Reference</i> .	X	NA	By default, the metadata block size is the same as the block size. For more information about block size, see the entry for -B <i>BlockSize</i> earlier in this table.
--mount-priority <i>Priority</i> to control the order in which the individual file systems are mounted at daemon startup or when one of the all keywords is specified on the mmmount command.	X	X	0
--perfilesset-quota to set the scope of user and group quota limit checks to the individual filesset level.	X	X	--noperfilesset-quota
--rapid-repair to keep track of incomplete replication on an individual file block basis (as opposed to the entire file).	NA	X	none
--version <i>VersionString</i> to enable only the file system features that are compatible with the specified release. See “Enabling file system features” on page 267.	X	NA	The default value is defined by the current committed function level for the cluster <code>minReleaseLevel</code> , for example 4.2.2.0 or other values, depending on the cluster.
--write-cache-threshold <i>HAWCThreshold</i> Specifies the maximum length in bytes of synchronous write requests that is initially buffered in the highly available write cache before being written back to primary storage.	X	NA	The default value is 0, which means that the option is disabled.
Notes: <ol style="list-style-type: none"> 1. X – indicates that the option is available on the command. 2. NA (not applicable) – indicates that the option is not available on the command. 			

Device name of the file system

File system names must be unique within a GPFS cluster. However, two different clusters can have two distinct file systems with the same name.

The device name of the file system does not need to be fully qualified. `fs0` is as acceptable as `/dev/fs0`. The name cannot be the same as an existing entry in `/dev`. The file system name must be no longer than 128 characters.

Note: If your cluster includes Windows nodes, the file system name must be no longer than 32 characters.

NFS V4 deny-write open lock

You can specify whether a deny-write open lock blocks writes, which is expected and required by NFS V4, Samba, and Windows.

For more information, see *Managing GPFS access control lists* in *IBM Spectrum Scale: Administration Guide*.

nfs4

Must be specified for file systems that support NFS V4 and file systems that are mounted on Windows. NFS4 is the default.

posix

Specified for file systems that support NFS V3 or file systems that are not NFS exported.

Allows NFS writes even in the presence of a deny-write open lock.

Disks for your file system

Disks must be defined as network shared disks (NSDs) before they can be added to a GPFS file system.

NSDs are created using the `mmcrnsd` command. Use the `mm1snsd -F` command to display a list of available NSDs. Use the `mmadddisk` to add the NSD after you create it.

See [“Disk considerations” on page 238](#).

Deciding how the file system is mounted

Specify when the file system is to be mounted:

yes

When the GPFS daemon starts. This is the default.

no

Manual mount.

automount

When the file system is first accessed.

This can be changed at a later time by using the `-A` option on the `mmchfs` command.

Considerations:

1. GPFS mount traffic may be lessened by using the automount feature to mount the file system when it is first accessed instead of at GPFS startup. Automatic mounts only produce additional control traffic at the point that the file system is first used by an application or user. Mounts at GPFS startup on the other hand produce additional control traffic at every GPFS startup. Thus startup of hundreds of nodes at once may be better served by using automatic mounts.
2. Automatic mounts will fail if the node does not have the operating system's automount support enabled for the file system.
3. When exporting file systems for NFS mounts, it may be useful to mount the file system when GPFS starts.

Block size

Choose the file system block size based on the projected workload of the file system and the type of storage that it uses.

[“General” on page 258](#)

[“Test actual performance with different block sizes” on page 259](#)

[“Factors that can affect performance” on page 259](#)

General

In a file system, a *block* is the largest contiguous amount of disk space that can be allocated to a file and also the largest amount of data that can be transferred in a single I/O operation. The block size determines the maximum size of a read request or write request that a file system sends to the I/O device driver. Blocks are composed of an integral number of *subblocks*, which are the smallest unit of contiguous disk space that can be allocated to a file. Files larger than one block are stored in some number of full blocks plus any subblocks that might be required after the last block to hold the remaining data. Files smaller than one block size are stored in one or more subblocks.

For information about setting block size and subblock size for a file system, see the descriptions of the `-B BlockSize` parameter and the `--metadata-block-size` parameter in help topic *mmcrfs command* in the *IBM Spectrum Scale: Command and Programming Reference*. Here are some general facts from those descriptions:

- 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.
- All the data blocks in a file system have the same block size and the same subblock size. Data blocks and subblocks in the system storage pool and those in user storage pools have the same sizes. An example of a valid block size and subblock size is a 4 MiB block with an 8 KiB subblock.
- All the metadata blocks in a file system have the same block size and the same subblock size. The metadata blocks and subblocks are set to the same sizes as data blocks and subblocks, unless the `--metadata-block-size` parameter is specified.

Note: The `--metadata-block-size` parameter that is used to specify a different metadata block size than the data block size is being deprecated. This option is no longer required to use for performance improvements for file systems with file system format 5.0.0 or later and it will be removed in a future release.

- If the system storage pool contains only `metadataOnly` NSDs, the metadata block can be set to a different size than the data block size with the `--metadata-block-size` parameter.

Note: This setting can result in a change in the data subblock size and in the number of subblocks in a data block, if the block size (`-B` parameter) is different from the `--metadata-block-size`. For an example, see Scenario 3 in a later bullet in this list.

- The data blocks and metadata blocks must have the same number of subblocks, even when the data block size and the metadata block size are different. See Scenario 3 in the next bullet.
- The number of subblocks per block is derived from the smallest block size of any storage pool in the file system, including the system metadata pool. Consider the following example scenarios:

Note: For a table of the valid block sizes and subblock sizes, see *mmcrfs command* in the *IBM Spectrum Scale: Command and Programming Reference*.

- Scenario 1: The file system is composed of a single system storage pool with all the NSD usage configured as `dataAndMetadata`. The file system block size is set with the `-B` parameter to 16MiB. As a result, the block size for both metadata and data blocks is 16 MiB. The metadata and data subblock size is 16 KiB.
- Scenario 2: The file system is composed of multiple storage pools with system storage pool NSD usage configured as `metadataOnly` and user storage pool NSD usage configured as `dataOnly`. The file system block size is set (`-B` parameter) to 16 MiB. The `--metadata-block-size` is also set to

16 MiB. As a result, the metadata and data block size is 16 MiB. The metadata and data subblock size is 16 KiB.

- Scenario 3: The file system is composed of multiple storage pools with the system storage pool NSD usage configured as `metadataOnly` and the user storage pool NSD usage configured as `dataOnly`. The file system block size is set (`-B` parameter) to 16 MiB, which has a subblock size of 16 KiB, but the `--metadata-block-size` is set to 1 MiB, which has a subblock size of 8 KiB. The number of subblocks across the pools of a file system needs to be the same and this is calculated based on the storage pool with smallest block size. In this case, the system pool has the smallest block size (1 MiB). The number of subblocks per block in the system storage pool is 128 (1 MiB block size / 8 KiB subblock size = 128 subblocks per block). The other storage pools inherit the 128-subblocks-per-block setting and their subblock size is recalculated based on 128 subblocks per block. In this case the subblock size of the user storage pool is recalculated as 128 KiB (16 MiB / 128 subblocks per block = 128 KiB subblock size)
- The block size cannot exceed the value of the cluster attribute `maxblocksize`, which can be set by the `mmchconfig` command.

Select a file system block size based on the workload and the type of storage. For a list of supported block sizes with their subblock sizes, see the description of the `-B BlockSize` parameter in help topic `mmcrfs command` in the *IBM Spectrum Scale: Command and Programming Reference*.



Attention: In IBM Spectrum Scale, the default block size of 4 MiB with an 8 KiB subblock size provides good sequential performance, makes efficient use of disk space, and provides good or similar performance for small files compared to the previous default block size of 256 KiB with 32 subblocks per block. It works well for the widest variety of workloads.

Test actual performance with different block sizes

The ideal file system block size can be determined by running performance tests with different file system block sizes using actual workloads or representative benchmarks that match the file sizes that you expect to use in production.

Factors that can affect performance

For more performance information, see the IBM Spectrum Scale white papers in the [Techdocs Library](http://www.ibm.com/support/techdocs/atsmastr.nsf/Web/WhitePapers) (www.ibm.com/support/techdocs/atsmastr.nsf/Web/WhitePapers).

RAID stripe size

The RAID stripe size is the size of the sequential block of data that a disk array writes to or reads from each storage volume (the block device corresponding to an NSD). For better performance, it is a good idea to set the file system block size to the same value as either the RAID stripe size or a multiple of the RAID stripe size. If the block size is not equal to or a multiple of the RAID stripe size, then the file system performance can be severely degraded, especially for write requests, because of the increase in read-modify-write operations that occur in the underlying hardware RAID controllers.

Note: The block size for IBM Spectrum Scale RAID that is implemented with `vdisk` is specifically designed for optimal behavior. For IBM Spectrum Scale RAID, the block size must be equal to the `vdisk` track size. For more information, see the online documentation available for IBM Spectrum Scale RAID in IBM Documentation.

File system size

For file systems larger than 100 TiB, it is a good idea to set the block size to at least 256 KiB. The default block size is 4 MiB in IBM Spectrum Scale. Generally larger block sizes provide better performance.

Large block size and page pool

For block sizes larger than the default size of 4 MiB, it is a good idea to increase the page pool size in proportion to the block size. The reason is that the efficiency of internal optimizations that rely

on caching file data in the GPFS page pool depends more on the number of blocks that are cached than on the amount data that is cached. A larger block size results in fewer cached blocks.

Variation in file size

For a file system that contain files of many different sizes, the file system delivers better overall performance from selecting a larger block size, 4 MiB or greater, rather than a smaller one. It is true that with a larger block size some space is wasted when a small file is written into a large subblock, because the unused space in the subblock cannot be written to with data from another file unless the block is freed.

However, the amount of waste in the general case is likely to be insignificant overall, because the smaller files occupy a smaller percentage of the storage space in the file system compared to the space occupied by the larger files (files on the order of GiBs).

Application I/O patterns

The effect of block size on file system performance greatly depends on the application I/O pattern:

- A larger block size is often beneficial for large sequential read and write workloads.
- A smaller block size can offer better performance for applications that do small random writes to sparse files or small random writes to large files that are subject to frequent snapshots.

Metadata performance

The choice of block size affects the performance of certain metadata operations, in particular, block allocation performance. The IBM Spectrum Scale block allocation map is stored in blocks, similar to regular files. When the block size is small:

- More blocks are required to store the same amount of data, which results in more work to allocate those blocks
- One block of allocation map data contains less information

Metadata-only system pool

The `--metadata-block-size` option on the `mmcrfs` command allows a different block size to be specified for the system storage pool, provided its usage is set to `metadataOnly`. Valid values are the same as the ones that are listed for the `-B` parameter.

Note: Setting the metadata block size to a different value than the data block size can have the effect of changing the data subblock size and the number of subblocks per data block. For more information see [Scenario 3](#) earlier in this help topic.

atime values

`atime` is a standard file attribute that represents the time when the file was last accessed.

The file attribute **`atime`** is updated locally in memory, but the value is not visible to other nodes until after the file is closed. To get an accurate value of **`atime`**, an application must call subroutine **`gpfs_stat`** or **`gpfs_fstat`**.

You can control how **`atime`** is updated with the **`-S`** option of the **`mmcrfs`** and **`mmchfs`** commands:

`-S no`

The **`atime`** attribute is updated whenever the file is read. This setting is the default if the minimum release level (**`minReleaseLevel`**) of the cluster is earlier than 5.0.0 when the file system is created.

`-S yes`

The **`atime`** attribute is not updated. The subroutines **`gpfs_stat`** and **`gpfs_fstat`** return the time that the file system was last mounted with **`-S no`**.

Note: This setting can cause a problem if you have a file management policy that depends on the **`ACCESS_TIME`** attribute. For more information, see the topic *Exceptions to Open Group technical standards* in the *IBM Spectrum Scale: Administration Guide*.

-S reltime

The **atime** attribute is updated whenever the file is read, but only if one of the following conditions is true:

- The current file access time (**atime**) is earlier than the file modification time (**mtime**).
- The current file access time (**atime**) is greater than the **atimeDeferredSeconds** attribute. For more information, see the topic *mmchconfig command* in the *IBM Spectrum Scale: Command and Programming Reference*.

This setting is the default if the minimum release level (**minReleaseLevel**) of the cluster is 5.0.0 or later when the file system is created.

You can temporarily override the value of the **-S** setting by specifying a mount option specific to IBM Spectrum Scale when you mount the file system. The mount option persists until the file system is unmounted. The following table shows how the mount options correspond to the **-S** settings:

Table 27. Correspondence between <i>mount</i> options and the -S option in <i>mmcrfs</i> and <i>mmchfs</i>		
-S option of mmcrfs and mmchfs commands.	Equivalent mount option; persists until file system is unmounted.	Effect – see topic text for details.
no	noreltime	Allow atime to be updated.
yes	noatime	Do not allow atime to be updated.
reltime	reltime	Allow atime to be updated if a condition is met.

For more information, see the topic *Mount options specific to IBM Spectrum Scale* in the *IBM Spectrum Scale: Administration Guide*.

mtime values

mtime is a standard file attribute that represents the time when the file was last modified.

The **-E** parameter controls when the **mtime** is updated. The default is **-E yes**, which results in standard interfaces including the `stat()` and `fstat()` calls reporting exact **mtime** values. Specifying **-E no** results in the `stat()` and `fstat()` calls reporting the **mtime** value available at the completion of the last sync period. This may result in the calls not always reporting the exact **mtime**. Setting **-E no** can affect backup operations, AFM, and AFM DR functions that rely on the last modified time or the operation of policies using the `MODIFICATION_TIME` file attribute.

For more information, see *Exceptions to the Open Group technical standards* in *IBM Spectrum Scale: Administration Guide*.

Block allocation map

GPFS has two different methods of allocating space in a file system. The **-j** parameter specifies the block allocation map type to use when creating a file system. The block allocation map type cannot be changed once the file system is created.

When allocating blocks for a given file, GPFS first uses a round-robin algorithm to spread the data across all of the disks in the file system. After a disk is selected, the location of the data block on the disk is determined by the block allocation map type.

The two types of allocation methods are **cluster** and **scatter**:

cluster

GPFS attempts to allocate blocks in clusters. Blocks that belong to a given file are kept next to each other within each cluster.

This allocation method provides better disk performance for some disk subsystems in relatively small installations. The benefits of clustered block allocation diminish when the number of nodes in the cluster or the number of disks in a file system increases, or when the file system free space becomes fragmented. The `cluster` allocation method is the default for GPFS clusters with eight or fewer nodes and for file systems with eight or fewer disks.

scatter

GPFS chooses the location of the blocks randomly.

This allocation method provides more consistent file system performance by averaging out performance variations due to block location (for many disk subsystems, the location of the data relative to the disk edge has a substantial effect on performance). This allocation method is appropriate in most cases and is the default for GPFS clusters with more than eight nodes or file systems with more than eight disks.

This parameter for a given file system is specified at file system creation by using the `-j` option on the `mmcrfs` command, or allowing it to default. This value *cannot* be changed after the file system has been created.

File system authorization

The type of authorization for the file system is specified on the `-k` option on the `mmcrfs` command or changed at a later time by using the `-k` option on the `mmchfs` command.

posix

Traditional GPFS access control lists (ACLs) only (NFS V4 and Windows ACLs are not allowed).

nfs4

Support for NFS V4 and Windows ACLs only. Users are not allowed to assign traditional ACLs to any file system objects.

all

Allows for the coexistence of POSIX, NFS V4, and Windows ACLs within a file system. This is the default.

If the file system will be used for the CES stack, `nfs4` is required.

Type of replication

You can control the type of replication that IBM Spectrum Scale uses.

Run the `mmchfs` command with the `-K` option set to the preferred type of replication. The `-K` option can have the following values:

always

Indicates that strict replication is enforced.

whenpossible

Strict replication is enforced if the disk configuration allows it. If the number of failure groups is insufficient, strict replication is not enforced. This is the default value.

no

Strict replication is not enforced. GPFS tries to create the needed number of replicas, but returns an `errno` of `EOK` if it can allocate at least one replica.

For more information on changing the attributes for replication, see the `mmchfs` command.

Internal log file

You can specify the internal log file size. Refer to [“GPFS recovery logs” on page 13](#) for additional information.

File system replication parameters

The metadata (inodes, directories, and indirect blocks) and data replication parameters are set at the file system level and apply to all files. They are initially set for the file system when issuing the `mmcrfs` command.

They can be changed for an existing file system using the `mmchfs` command. When the replication parameters are changed, files created after the change are affected. To apply the new replication values to existing files in a file system, issue the `mmrestripefs` command.

Metadata and data replication are specified independently. Each has a default replication factor of 1 (no replication) and a maximum replication factor with a default of 2. Although replication of metadata is less costly in terms of disk space than replication of file data, excessive replication of metadata also affects GPFS efficiency because all metadata replicas must be written. In general, more replication uses more space.

Default metadata replicas

The default number of copies of metadata for all files in the file system may be specified at file system creation by using the `-m` option on the `mmcrfs` command or changed at a later time by using the `-m` option on the `mmchfs` command.

This value must be equal to or less than *MaxMetadataReplicas*, and cannot exceed the number of failure groups with disks that can store metadata. The allowable values are 1, 2, or 3, with a default of 1 (no replication).

Related concepts

Maximum metadata replicas

The maximum number of copies of metadata for all files in the file system can be specified at file system creation by using the `-M` option on the `mmcrfs` command.

Default data replicas

The default replication factor for data blocks may be specified at file system creation by using the `-r` option on the `mmcrfs` command or changed at a later time by using the `-r` option on the `mmchfs` command.

Maximum data replicas

The maximum number of copies of data blocks for a file can be specified at file system creation by using the `-R` option on the `mmcrfs` command. The default is 2.

Maximum metadata replicas

The maximum number of copies of metadata for all files in the file system can be specified at file system creation by using the `-M` option on the `mmcrfs` command.

The default is 2. The allowable values are 1, 2, or 3, but it cannot be less than the value of *DefaultMetadataReplicas*. This value cannot be changed after the file system is created.

Related concepts

Default metadata replicas

The default number of copies of metadata for all files in the file system may be specified at file system creation by using the `-m` option on the `mmcrfs` command or changed at a later time by using the `-m` option on the `mmchfs` command.

Default data replicas

The default replication factor for data blocks may be specified at file system creation by using the `-r` option on the `mmcrfs` command or changed at a later time by using the `-r` option on the `mmchfs` command.

Maximum data replicas

The maximum number of copies of data blocks for a file can be specified at file system creation by using the `-R` option on the `mmcrfs` command. The default is 2.

Default data replicas

The default replication factor for data blocks may be specified at file system creation by using the `-r` option on the `mmcrfs` command or changed at a later time by using the `-r` option on the `mmchfs` command.

This value must be equal to or less than *MaxDataReplicas*, and the value cannot exceed the number of failure groups with disks that can store data. The allowable values are 1, 2, and 3, with a default of 1 (no replication).

If you want to change the data replication factor for the entire file system, the data disk in each storage pool must have a number of failure groups equal to or greater than the replication factor. For example, you will get a failure with error messages if you try to change the replication factor for a file system to 2 but the storage pool has only one failure group.

Related concepts

Default metadata replicas

The default number of copies of metadata for all files in the file system may be specified at file system creation by using the `-m` option on the `mmcrfs` command or changed at a later time by using the `-m` option on the `mmchfs` command.

Maximum metadata replicas

The maximum number of copies of metadata for all files in the file system can be specified at file system creation by using the `-M` option on the `mmcrfs` command.

Maximum data replicas

The maximum number of copies of data blocks for a file can be specified at file system creation by using the `-R` option on the `mmcrfs` command. The default is 2.

Maximum data replicas

The maximum number of copies of data blocks for a file can be specified at file system creation by using the `-R` option on the `mmcrfs` command. The default is 2.

The allowable values are 1, 2, and 3, but cannot be less than the value of *DefaultDataReplicas*. This value cannot be changed after the file system is created.

Related concepts

Default metadata replicas

The default number of copies of metadata for all files in the file system may be specified at file system creation by using the `-m` option on the `mmcrfs` command or changed at a later time by using the `-m` option on the `mmchfs` command.

Maximum metadata replicas

The maximum number of copies of metadata for all files in the file system can be specified at file system creation by using the `-M` option on the `mmcrfs` command.

Default data replicas

The default replication factor for data blocks may be specified at file system creation by using the `-r` option on the `mmcrfs` command or changed at a later time by using the `-r` option on the `mmchfs` command.

Number of nodes mounting the file system

The estimated number of nodes that will mount the file system may be specified at file system creation by using the `-n` option on the `mmcrfs` command or allowed to default to 32.

When creating a GPFS file system, over-estimate the number of nodes that will mount the file system. This input is used in the creation of GPFS data structures that are essential for achieving the maximum degree of parallelism in file system operations (see [“GPFS architecture” on page 8](#)). Although a larger estimate consumes a bit more memory, insufficient allocation of these data structures can limit the ability to process certain parallel requests efficiently, such as the allotment of disk space to a file. If you cannot

predict the number of nodes, allow the default value to be applied. Specify a larger number if you expect to add nodes, but avoid wildly overestimating as this can affect buffer operations.

You can change the number of nodes using the `-n` option on the `mmchfs` command. Changing this value affects storage pools created after the value was set; so, for example, if you need to increase this value on a storage pool, you could change the value, create a new storage pool, and migrate the data from one pool to the other.

Windows drive letter

In a Windows environment, you must associate a drive letter with a file system before it can be mounted. The drive letter can be specified and changed with the `-t` option of the `mmcrfs` and `mmchfs` commands. GPFS does not assign a default drive letter when one is not specified.

The number of available drive letters restricts the number of file systems that can be mounted on Windows.

Note: Certain applications give special meaning to drive letters A :, B :, and C :, which could cause problems if they are assigned to a GPFS file system.

Mountpoint directory

Every GPFS file system has a default mount point associated with it. This mount point can be specified and changed with the `-T` option of the `mmcrfs` and `mmchfs` commands.

If you do not specify a mount point when you create the file system, GPFS will set the default mount point to `/gpfs/DeviceName`.

Assign mount command options

Options may be passed to the file system mount command using the `-o` option on the `mmchfs` command.

In particular, you can choose the option to perform quota activation automatically when a file system is mounted.

Enabling quotas

The IBM Spectrum Scale quota system can help you control file system usage.

Quotas can be defined for individual users, groups of users, or filesets. Quotas can be set on the total number of files and on the total amount of data space that is used.

If data replication is configured, be sure to include the size of the replicated data in your calculation of the proper size for a quota limit. For more information, see the topic *Listing quotas* in the *IBM Spectrum Scale: Administration Guide*.

To have IBM Spectrum Scale automatically enable quotas when a file system is mounted, choose one of the following options:

- When you create the file system, issue the `mmcrfs` command with the `-Q` option included.
- After the file system is created, issue the `mmchfs` command with the `-Q` option included.

After the file system is mounted, you can set quota values by issuing the `mmedquota` command and activate quotas by issuing the `mmquotaon` command. By default, quota values are not automatically activated when they are set.

Quota levels are defined at three limits that you can set with the `mmedquota` and `mmdefedquota` commands:

Soft limit

Defines levels of disk space and files below which the user, group, or fileset can safely operate.

Specified in units of KiB (k or K), MiB (m or M), or GiB (g or G). If no suffix is provided, the number is assumed to be in bytes.

Hard limit

Defines the maximum amount of disk space and number of files that the user, group, or fileset can accumulate.

Specified in units of KiB (k or K), MiB (m or M), or GiB (g or G). If no suffix is provided, the number is assumed to be in bytes.

Grace period

Allows the user, group, or fileset to exceed the soft limit for a specified period of time. The default period is one week. If usage is not reduced to a level below the soft limit during that time, the quota system interprets the soft limit as the hard limit and no further allocation is allowed. The user, group, or fileset can reset this condition by reducing usage enough to fall below the soft limit; or the administrator can increase the quota levels with the `mmedquota` or `mmdefedquota`.

The use of SMB, NFS, and object protocols affects the operation of quotas. For more information, see *Implications of quotas for different protocols* in the *IBM Spectrum Scale: Administration Guide*.

Default quotas

Applying default quotas provides all new users, groups of users, or filesets with established minimum quota limits. If default quota values are not enabled, new users, new groups, or new filesets have a quota value of zero, which establishes no limit to the amount of space that can be used.

Default quota limits can be set or changed only if the `-Q yes` option is in effect for the file system. To set default quotas at the fileset level, the `--perfileset-quota` option must also be in effect. 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 `mmllsfs` command to display the current settings of these quota options. Default quotas may then be enabled by issuing the `mmdefquotaon` command. Default values are established by issuing the `mmdefedquota` command.

Enabling DMAPI

Whether or not the file system can be monitored and managed by the GPFS Data Management API (DMAPI) may be specified at file system creation by using the `-z` option on the `mmcrfs` command or changed at a later time by using the `-z` option on the `mmchfs` command.

The default is *not* to enable DMAPI for the file system.

For more information about DMAPI for GPFS, see *IBM Spectrum Scale: Command and Programming Reference*.

Verifying disk usage

The `-v` option controls whether the `mmcrfs` command checks whether the specified disks can safely be added to the file system.

The default (`-v yes`) is to perform the check and fail the command if any of the disks appear to belong to some other GPFS file system. You should override the default behavior and specify `-v no` only if the `mmcrfs` command rejects the disks and you are certain that *all* of the disks indeed do not belong to an active GPFS file system. An example for an appropriate use of `-v no` is the case where an `mmcrfs` command is interrupted for some reason and you are reissuing the command. Another example would be if you are reusing disks from an old GPFS file system that was not formally destroyed with the `mmde1fs` command.

Important: Using `mmcrfs -v no` on a disk that already belongs to a file system will corrupt that file system.

Changing the file system format to the latest level

You can change the file system format to the latest format supported by the currently-installed level of GPFS by issuing the `mmchfs` command with the `-V full` option or the `-V compat` option.

The `full` option enables all new functionality that requires different on-disk data structures. This may cause the file system to become permanently incompatible with earlier releases of GPFS. The `compat`

option enables only changes that are backward compatible with the previous GPFS release. If all GPFS nodes that are accessing a file system (both local and remote) are running the latest level, then it is safe to use `full` option. Certain features may require you to run the `mmigratefs` command to enable them.

For more information, see *File system format changes between versions of GPFS* in *IBM Spectrum Scale: Administration Guide*.

Enabling file system features

By default, new file systems are created with all currently available features enabled.

Since this may prevent clusters that are running earlier GPFS releases from accessing the file system, you can enable only the file system features that are compatible with the specified release by issuing the `mmcrfs` command with the `--version Version` option. For more information, see *mmcrfs command* in *IBM Spectrum Scale: Command and Programming Reference*.

Specifying whether the `df` command will report numbers based on quotas for the fileset

You can specify (when quotas are enforced for a fileset) whether the `df` command will report numbers based on the quotas for the fileset and not for the total file system. If `ignoreReplicationOnStatfs` is enabled, `df` command on fileset output ignores data replication factor.

To do so, use the `--filesetdf` | `--nofilesetdf` option on either the `mmchfs` command or the `mmcrfs` command. This option affects the `df` command behavior only on Linux nodes.

If quota is disabled and `filesetdf` is enabled in IBM Spectrum Scale 5.1.1 or later with file system version 5.1.1 or later, then the **`df`** command reports inode space capacity and inode usage at the independent fileset-level. However, the **`df`** command reports the block space at the file system-level because the block space is shared with the whole file system.

Note: If quota is enabled, then no behavior change for **`df`** command with **`filesetdf`** enabled, regardless of the cluster and file system versions.

For more information, see *mmchfs command* and *mmcrfs command* in *IBM Spectrum Scale: Command and Programming Reference*.

Specifying the maximum number of files that can be created

The maximum number of files that can be created can be specified by using the `--inode-limit` option on the `mmcrfs` command and the `mmchfs` command.

Allowable values, which range from the current number of created inodes (determined by issuing the `mmddf` command with the `-F` option) through the maximum number of files that are supported, are constrained by the formula:

```
maximum number of files = (total file system space) / (inode size + subblock size)
```

You can determine the inode size (`-i`) and subblock size (`-f`) of a file system by running the `mmllsfs` command. The maximum number of files in a file system may be specified at file system creation by using the `--inode-limit` option on the `mmcrfs` command, or it may be increased at a later time by using `--inode-limit` on the `mmchfs` command. This value defaults to the size of the file system at creation divided by 1 MB and cannot exceed the architectural limit. When a file system is created, 4084 inodes are used by default; these inodes are used by GPFS for internal system files.

For more information, see *mmcrfs command* and *mmchfs command* in *IBM Spectrum Scale: Command and Programming Reference*.

The `--inode-limit` option applies only to the root fileset. Preallocated inodes created using the **`mmcrfs`** command are allocated only to the root fileset and these inodes cannot be deleted or moved to another independent fileset. When there are multiple inode spaces, use the `--inode-limit` option

of the `mmchfileset` command to alter the inode limits of independent filesets. The `mmchfileset` command can also be used to modify the root inode space. The `--inode-limit` option of the `mmlsfs` command shows the sum of all inode spaces.

Inodes are allocated when they are used. When a file is deleted, the inode is reused, but inodes are never deallocated. When setting the maximum number of inodes in a file system, there is the option to preallocate inodes. However, in most cases there is no need to preallocate inodes because, by default, inodes are allocated in sets as needed. If you do decide to preallocate inodes, be careful not to preallocate more inodes than will be used; otherwise, the allocated inodes will unnecessarily consume metadata space that cannot be reclaimed, and it might result in excessive memory use associated with the objects allocated with the inode segments.

These options limit the maximum number of files that may actively exist within a file system. However, the maximum number of files in the file system may be restricted further by GPFS so the control structures associated with each file do not consume all of the file system space.

Further considerations when managing inodes:

1. For file systems that are supporting parallel file creates, as the total number of free inodes drops below 5% of the total number of inodes, there is the potential for slowdown in file system access. Take this into consideration when creating or changing your file system. Use the `mmdf` command to display the number of free inodes.
2. Excessively increasing the value for the maximum number of files may cause the allocation of too much disk space for control structures.

Controlling the order in which file systems are mounted

You can control the order in which the individual file systems are mounted at daemon startup or when using the `mmmount` command with one of the `all` keywords specified for the file system.

For more information, see *mmlsfs command* and *mmmount command* in *IBM Spectrum Scale: Command and Programming Reference*.

To do so, use the `--mount-priority` *Priority* option on the `mmcrfs`, the `mmchfs`, or the `mmremotefs` command.

The shared root file system for protocols must be mounted before other file systems that will be used to export protocol data.

A sample file system creation

To create a file system called `gpfs2` with the following properties:

- The disks for the file system that are listed in the file `/tmp/gpfs2dsk`
- Automatically mount the file system when the GPFS daemon starts (`-A yes`)
- Use a block size of 256 KB (`-B 256K`)
- Expect to mount it on 32 nodes (`-n 32`)
- Set both the default metadata replication and the maximum metadata replication to two (`-m 2 -M 2`)
- Set the default data replication to one and the maximum data replication to two (`-r 1 -R 2`)
- Use a default mount point of `/gpfs2` (`-T /gpfs2`)

Enter:

```
mmcrfs /dev/gpfs2 -F /tmp/gpfs2dsk -A yes -B 256K -n 32 -m 2 -M 2 -r 1 -R 2 -T /gpfs2
```

The system displays information similar to:

```
The following disks of gpfs2 will be formatted on node k194p03.tes.nnn.com:
hd25n09: size 17796014 KB
hd24n09: size 17796014 KB
hd23n09: size 17796014 KB
Formatting file system ...
Disks up to size 59 GB can be added to storage pool system.
```

```

Creating Inode File
 56 % complete on Mon Mar  3 15:10:08 2014
100 % complete on Mon Mar  3 15:10:11 2014
Creating Allocation Maps
Clearing Inode Allocation Map
Clearing Block Allocation Map
 44 % complete on Mon Mar  3 15:11:32 2014
 90 % complete on Mon Mar  3 15:11:37 2014
100 % complete on Mon Mar  3 15:11:38 2014
Completed creation of file system /dev/gpfs2.
mmcrfs: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.

```

To confirm the file system configuration, issue the command:

```
mmlsfs gpfs2
```

The system displays information similar to:

flag	value	description
-f	262144	Minimum fragment (subblock) size in bytes
-i	512	Inode size in bytes
-I	32768	Indirect block size in bytes
-m	2	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	scatter	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
--perfilesset-quota	yes	Per-filesset quota enforcement
--filesetdf	yes	Filesset df enabled?
-V	15.01 (4.2.0.0)	File system version
--create-time	Wed Jan 18 17:22:25 2017	File system creation time
-z	no	Is DMAPI enabled?
-L	262144	Logfile size
-E	yes	Exact mtime mount option
-S	yes	Suppress atime mount option
-K	whenpossible	Strict replica allocation option
--fastea	yes	Fast external attributes enabled?
--encryption	no	Encryption enabled?
--inode-limit	2015232	Maximum number of inodes
--log-replicas	0	Number of log replicas (max 2)
--is4KAligned	yes	is4KAligned?
--rapid-repair	yes	rapidRepair enabled?
--write-cache-threshold	65536	HAWC Threshold (max 65536)
-P	system	Disk storage pools in file system
-d	gpfs1001nsd;gpfs1002nsd	Disks in file system
-A	yes	Automatic mount option
-o	none	Additional mount options
-T	/gpfs2	Default mount point
--mount-priority	0	Mount priority

For more information, see *mmcrfs* command and *mmlsfs* command in *IBM Spectrum Scale: Command and Programming Reference*.

Backup considerations for using IBM Spectrum Protect

If you are planning to use IBM Spectrum Protect to back up IBM Spectrum Scale file systems, some considerations apply.

Considerations for provisioning IBM Spectrum Protect servers to handle backup of IBM Spectrum Scale file systems

When backing up IBM Spectrum Scale file systems, the IBM Spectrum Protect server must be configured with adequate resources to handle a much larger load of client data, sessions, and mount points than typically encountered with single-user file systems such as workstations.

For cluster file system backup, a system architect must plan for multiple nodes (storage hosts) sending data in multiple streams, at high data rates to the server. Therefore, the IBM Spectrum Protect server must be equipped with the following:

- High bandwidth network connections to the IBM Spectrum Scale cluster
- Sufficient disk space organized into an IBM Spectrum Protect storage pool to be allocated to backup data from the cluster file system
- Sufficient number of mount points (or infinite for a disk class pool) accorded to the proxy client used for the cluster file system
- Sufficient number of IBM Spectrum Protect sessions permitted to the proxy client used for the cluster file system
- Sufficient number of physical tape drives to offload the backup data once it arrives into the disk pool

Related concepts

IBM Spectrum Protect data storage model

When using IBM Spectrum Protect for backing up IBM Spectrum Scale file systems, the IBM Spectrum Protect data storage architecture and its implications need to be considered.

How to identify backup and migration candidates

When using IBM Spectrum Protect for backing up IBM Spectrum Scale file systems, there are several choices for the method used to identify backup and migration candidates.

Comparison of snapshot based backups and backups from live system

Backing up large file systems can take many hours or even days. When using the IBM Spectrum Scale command **mmbackup**, time is needed for the following steps.

File system and fileset backups with IBM Spectrum Protect

mmbackup supports backup from independent filesets in addition to backup of the whole file system. Independent filesets have similar capabilities as a file system in terms of quota management, dependent filesets, snapshots, and having their own inode space.

Considerations for using fileset backup with IBM Spectrum Protect

IBM Spectrum Protect stores backup data indexed by file or directory object path names in its database. However, file system objects are identified for IBM Spectrum Protect by their object ID extended attributes. In both cases, IBM Spectrum Protect is not aware of the fileset entity. Therefore, the fileset configuration of an IBM Spectrum Scale cluster is neither backed up nor can it be restored by IBM Spectrum Protect alone, unless separate action is taken.

Considerations for backing up file systems that are managed with IBM Spectrum Protect for Space Management

When IBM Spectrum Protect for Space Management is used to migrate data to secondary storage pools (for example, tape volumes), this migrated data might need to be recalled to disk if the file becomes a backup candidate due to certain changes a user may apply to it.

IBM Spectrum Protect data storage model

When using IBM Spectrum Protect for backing up IBM Spectrum Scale file systems, the IBM Spectrum Protect data storage architecture and its implications need to be considered.

In IBM Spectrum Scale, the following kinds of data are stored on disk:

- The file data that is the content stored in a file.
- The file metadata that includes all attributes related to the file. For example:
 - Create, access, and modify times
 - Size of the file, size occupied in file system, and number of blocks used
 - Inode information, owner user id, owning group id, and mode bits
 - POSIX rights or access control lists (ACLs)
 - Flags to indicate whether the file is immutable or mutable, read only, or append only
 - Extended attributes (EAs)

In IBM Spectrum Protect, the same file data and metadata is stored but the method of storing this data differs. The file content is stored in an IBM Spectrum Protect storage pool such as on disk or on tape while some of the metadata is stored in the IBM Spectrum Protect database. The primary reason for storing metadata in the IBM Spectrum Protect database is to provide fast access to information useful for backup requests.

However, not all metadata is stored in the IBM Spectrum Protect database. Access control lists (ACLs) and extended attributes (EAs) are stored with the file content in the storage pool (media depends on storage pool type). This has the following implications:

- When the ACL or EA of a file changes then the next backup job backs up the whole file again. This occurs because the file content, ACL, and EA are stored together in the IBM Spectrum Protect data pool, for example on tape and they need to be updated as one entity.

Note: ACLs are inherited in the IBM Spectrum Scale file system. Therefore, an ACL on a top-level directory object can be inherited to all the descendant objects. ACL changes to a top-level directory object are therefore propagated down through the object tree hierarchy, rippling the change through all objects that inherited the original ACL. The number of files to be backed up increases even though nothing else in these files has changed. A surprisingly large backup workload can be induced by a seemingly small change to an ACL to a top level directory object.

When IBM Spectrum Protect for Space Management capability is enabled, an ACL change such as this occurs to objects that are currently migrated to offline storage as well. These files will then need to be recalled during the next backup cycle to enable the updated ACL to be stored with the file data once again in their IBM Spectrum Protect backup storage pool.

- Renaming of a file also leads to a backup of the whole file because the IBM Spectrum Protect database is indexed by file object path name.

You can use the following approaches to mitigate the size of a backup workload when widely inherited ACLs are likely to be changed frequently.

- Avoid renaming directories that are close to the file system root.
- Avoid ACL and EA changes in migrated files as much as possible.
- Consider using the `skipacl` or `skipaclupdatecheck` options of the IBM Spectrum Protect client.

Important: Be certain to note the implications of using these options by referring to *Clients options reference* in the *Backup-archive Client options and commands* section of IBM Spectrum Protect documentation.

Note: Using the `skipacl` option also omits EAs from the backup data store in the IBM Spectrum Protect backup pool. 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 Spectrum Protect restore operation. If you are using this approach, ensure that the ACL is restored either manually or automatically, by inheritance, to avoid an unauthorized user getting access to a file or a directory after it is restored.

Related concepts

[Considerations for provisioning IBM Spectrum Protect servers to handle backup of IBM Spectrum Scale file systems](#)

When backing up IBM Spectrum Scale file systems, the IBM Spectrum Protect server must be configured with adequate resources to handle a much larger load of client data, sessions, and mount points than typically encountered with single-user file systems such as workstations.

[How to identify backup and migration candidates](#)

When using IBM Spectrum Protect for backing up IBM Spectrum Scale file systems, there are several choices for the method used to identify backup and migration candidates.

[Comparison of snapshot based backups and backups from live system](#)

Backing up large file systems can take many hours or even days. When using the IBM Spectrum Scale command **mmbackup**, time is needed for the following steps.

[File system and fileset backups with IBM Spectrum Protect](#)

mmbackup supports backup from independent filesets in addition to backup of the whole file system. Independent filesets have similar capabilities as a file system in terms of quota management, dependent filesets, snapshots, and having their own inode space.

[Considerations for using fileset backup with IBM Spectrum Protect](#)

IBM Spectrum Protect stores backup data indexed by file or directory object path names in its database. However, file system objects are identified for IBM Spectrum Protect by their object ID extended attributes. In both cases, IBM Spectrum Protect is not aware of the fileset entity. Therefore, the fileset configuration of an IBM Spectrum Scale cluster is neither backed up nor can it be restored by IBM Spectrum Protect alone, unless separate action is taken.

[Considerations for backing up file systems that are managed with IBM Spectrum Protect for Space Management](#)

When IBM Spectrum Protect for Space Management is used to migrate data to secondary storage pools (for example, tape volumes), this migrated data might need to be recalled to disk if the file becomes a backup candidate due to certain changes a user may apply to it.

How to identify backup and migration candidates

When using IBM Spectrum Protect for backing up IBM Spectrum Scale file systems, there are several choices for the method used to identify backup and migration candidates.

Limitations when using IBM Spectrum Protect Backup-Archive client to identify backup candidates

Using IBM Spectrum Protect Backup-Archive client to traverse IBM Spectrum Scale file systems to identify backup candidates does not scale well. For this reason, using the IBM Spectrum Scale **mmapplypolicy** engine is preferable because it is much faster to scan the file system for identifying backup candidates than traversing the file system.

Therefore, for processing backups on larger file systems, use the IBM Spectrum Scale command **mmbackup** instead of using the IBM Spectrum Protect Backup-Archive client commands such as **dsmsc expire** or **dsmsc selective** or **dsmsc incremental** directly. Using the **mmbackup** command also provides the following benefits:

- Backup activities are run in parallel by using multiple IBM Spectrum Scale cluster nodes to send backup data in parallel to the IBM Spectrum Protect server.

- **mmbbackup** creates a local shadow of the IBM Spectrum Protect database in the file system and uses it along with the policy engine to identify candidate files for backup. The IBM Spectrum Protect server does not need to be queried for this information saving time when calculating the backup candidate list.
 - **mmbbackup** and its use of the policy engine can select candidates faster than the **dsmc** progressive incremental operation that is bounded by walk of the file system using the POSIX directory and file status reading functions.
 - Using **dsmc selective** with lists generated by **mmbbackup** is also faster than using **dsmc incremental** even with similar lists generated by **mmbbackup**.

Note: It is recommended that scheduled backups of an IBM Spectrum Scale file system use **mmbbackup** because **mmbbackup** does not actively query the IBM Spectrum Protect server to calculate backup candidates. However, events such as file space deletion or file deletion executed on IBM Spectrum Protect server are not recognized until the user triggers a synchronization between the **mmbbackup** shadow database and the IBM Spectrum Protect database.

The following table contains a detailed comparison of **mmbbackup** and IBM Spectrum Protect Backup-Archive client backup commands:

<i>Table 28. Comparison of mmbbackup and IBM Spectrum Protect Backup-Archive client backup commands</i>		
	IBM Spectrum Scale policy-driven backup (mmbbackup)	IBM Spectrum Protect progressive incremental backup (dsmc incremental)
Detects changes in files and sends a new copy of the file to the server.	Yes	Yes
Detects changes in metadata and updates the file metadata on the server or sends a new copy of the file to the server (for ACL/EA changes).	Yes	Yes
Detects directory move, copy, or rename functions, and sends a new copy of the file to the server.	Yes	Yes
Detects local file deletion and expires the file on the server.	Yes	Yes
Detects IBM Spectrum Protect file space deletion or node/policy changes, and sends a new copy of the file to the server.	No*	Yes
Detects file deletion from the IBM Spectrum Protect server and sends a new copy of the file to the server.	No*	Yes
Detects additions of new exclude rules and expires the file on the server.	Yes	Yes
Detects policy changes made to include rules and rebinds the file to the new storage pool.	No**	Yes
Detects copy mode and copy frequency configuration changes.	No*	Yes

Table 28. Comparison of **mmbackup** and IBM Spectrum Protect Backup-Archive client backup commands (continued)

	IBM Spectrum Scale policy-driven backup (mmbackup)	IBM Spectrum Protect progressive incremental backup (dsmc incremental)
Detects migration state changes (IBM Spectrum Protect for Space Management) and updates the server object.	Yes	Yes
Detects that a file wasn't processed successfully during a backup operation and attempts again at the next backup.	Yes	Yes
Supports IBM Spectrum Protect Virtual Mount Points to divide a file system into smaller segments to reduce database size and contention.	No	Yes
* The mmbackup command queries the IBM Spectrum Protect server only once at the time of the first backup. Changes that are performed on the IBM Spectrum Protect server by using the IBM Spectrum Protect administrative client cannot be detected by mmbackup processing. You must rebuild the mmbackup shadow database if the IBM Spectrum Protect server file space changes.		
** IBM Spectrum Protect includes rules with associated management class bindings that cannot be detected by mmbackup processing. Therefore, mmbackup processing does not rebind a file if a management class changes include rules.		

If you use IBM Spectrum Protect Backup-Archive client backup commands on file systems that are otherwise handled by using **mmbackup**, the shadow database maintained by **mmbackup** loses its synchronization with the IBM Spectrum Protect inventory. In such cases, you need to resynchronize with the IBM Spectrum Protect server which will inform **mmbackup** of the recent backup activities conducted with the **dsmc** command. Resynchronization might be a very time-consuming activity for large file systems with a high number of backed up items. To avoid these scenarios, use the **mmbackup** command only.

If you have used **dsmc selective** or **dsmc incremental** since starting to use **mmbackup** and need to manually trigger a synchronization between the **mmbackup** maintained shadow database and the IBM Spectrum Protect server:

- Use the **mmbackup --rebuild** if you need to do a synchronization only.
- Use the **mmbackup -q** if you need to do a synchronization followed by a backup of the corresponding file system.

Using the IBM Spectrum Protect for Space Management client to identify migration candidates

Using IBM Spectrum Protect for Space Management clients for traversing IBM Spectrum Scale file system to identify migration candidates does not scale well. The IBM Spectrum Protect **automigration** daemons consume space in the file system and also consume CPU resources. They do not have access to the internal structures of the file system in the way that the IBM Spectrum Scale **mmapplypolicy** command does, and so they cannot scale. Use the following steps instead:

1. Set the following environment variable before the installation of the IBM Spectrum Protect for Space Management client to prevent the **automigration** daemons from starting during the installation:

```
export HSMINSTALLMODE=SCOUTFREE
```

It is recommended to place this setting into the profile file of the root user.

2. Add the following option to your IBM Spectrum Protect client configuration file, `dsm.opt`, to prevent the **automigration** daemons from starting after every system reboot:

`HSMDISABLEAUTOMIGDAEMONS YES`
3. Add the following option to your IBM Spectrum Protect client configuration file, `dsm.opt`, to ensure that the object ID is added to the inode, so that the file list based reconciliation (two-way-orphan-check) can be used:

`HSMEXTOBJidattr YES`
4. While the **automigration** daemons are disabled, changes such as removal of migrated files are not automatically propagated to the IBM Spectrum Protect server. For housekeeping purposes, you must run the IBM Spectrum Protect reconciliation either manually or in a scheduled manner. For more information, see *Reconciling by using a GPFS policy* in IBM Spectrum Protect for Space Management documentation.

Related concepts

Considerations for provisioning IBM Spectrum Protect servers to handle backup of IBM Spectrum Scale file systems

When backing up IBM Spectrum Scale file systems, the IBM Spectrum Protect server must be configured with adequate resources to handle a much larger load of client data, sessions, and mount points than typically encountered with single-user file systems such as workstations.

IBM Spectrum Protect data storage model

When using IBM Spectrum Protect for backing up IBM Spectrum Scale file systems, the IBM Spectrum Protect data storage architecture and its implications need to be considered.

Comparison of snapshot based backups and backups from live system

Backing up large file systems can take many hours or even days. When using the IBM Spectrum Scale command **mmbackup**, time is needed for the following steps.

File system and fileset backups with IBM Spectrum Protect

mmbackup supports backup from independent filesets in addition to backup of the whole file system. Independent filesets have similar capabilities as a file system in terms of quota management, dependent filesets, snapshots, and having their own inode space.

Considerations for using fileset backup with IBM Spectrum Protect

IBM Spectrum Protect stores backup data indexed by file or directory object path names in its database. However, file system objects are identified for IBM Spectrum Protect by their object ID extended attributes. In both cases, IBM Spectrum Protect is not aware of the fileset entity. Therefore, the fileset configuration of an IBM Spectrum Scale cluster is neither backed up nor can it be restored by IBM Spectrum Protect alone, unless separate action is taken.

Considerations for backing up file systems that are managed with IBM Spectrum Protect for Space Management

When IBM Spectrum Protect for Space Management is used to migrate data to secondary storage pools (for example, tape volumes), this migrated data might need to be recalled to disk if the file becomes a backup candidate due to certain changes a user may apply to it.

Comparison of snapshot based backups and backups from live system

Backing up large file systems can take many hours or even days. When using the IBM Spectrum Scale command **mmbackup**, time is needed for the following steps.

- Scanning the system to identify the objects that need to be backed up or expired.
- Expiring all objects removed from the system.
- Backing up all new or changed objects.

If the backup is run on the live file system while it is active, objects selected for the backup job can be backed up at different points in time. This can lead to issues when temporary or transient files that were present during the scan time are removed by the time the backup command tries to send them to the IBM Spectrum Protect server. The attempt to back up a file that is removed fails and the need to back this object up is still recorded in the shadow database.

Instead of backing up from a live file system, an alternative is to use snapshot based backups. Using the snapshot adds additional actions of creating or reusing a snapshot and removing it when the overall backup process completes. However, this approach provides several advantages because a snapshot is a point in time view of the file system that is read-only and it can be used for backing up the file system for as long as is necessary to complete the backup job. The advantages of following this approach are as follows:

- Transient or temporary files are backed up, provided they existed at the time the snapshot was taken.
- Protection against failures to back up due to server-side faults such as the IBM Spectrum Protect server running out of space. For example, if the database or storage pool becomes full or if the IBM Spectrum Protect server crashes, etc. In this case, a retry of the backup is possible for the point in time when the snapshot has been taken with no loss of function or backup.
- Retention of a backup within the system. Snapshots can be kept for a period of time providing an online backup copy of all files. This can protect against accidental deletions or modifications, and can be used to retrieve an earlier version of a file, etc.
- A means to fulfill a data protection policy even if the backup activity to IBM Spectrum Protect exceeds the nominal time window allotted. The snapshot can be kept for several days until backups are complete, and multiple snapshots can be kept until backup completes for each of them.

IBM Spectrum Scale provides the capability to create snapshots for a complete file system, known as global snapshots, and for independent filesets, known as fileset snapshots.

While snapshot based backups provide several advantages, the following considerations apply when using the snapshot capability:

- Snapshots might consume space; usually a snapshot used for backup is removed shortly after the backup operation finishes. Long lived snapshots retain their copy of the data blocks, taken from the file system's pool of free blocks. As they age, the snapshots consume more data blocks owing to the changes made in the read or write view of the file system.
- Snapshot deletion can take time depending on the number of changes that need to be handled while removing the snapshot. In general, the older a snapshot is, the more work it will require to delete it.
- Special consideration for use of IBM Spectrum Protect for Space Management:
 - When a migrated file stub that is already part of a snapshot is removed, a recall is initiated to keep the snapshot consistent. This is required because removal of the stub invalidates the offline references to the stored data. The recall is to fill blocks on disk and assign them to the snapshot view. Once the stub is removed from the file system and a reconcile process removes this file from the Space Management pool on the IBM Spectrum Protect server, there are no longer any references to the file data except the snapshot copy.

Related concepts

Considerations for provisioning IBM Spectrum Protect servers to handle backup of IBM Spectrum Scale file systems

When backing up IBM Spectrum Scale file systems, the IBM Spectrum Protect server must be configured with adequate resources to handle a much larger load of client data, sessions, and mount points than typically encountered with single-user file systems such as workstations.

IBM Spectrum Protect data storage model

When using IBM Spectrum Protect for backing up IBM Spectrum Scale file systems, the IBM Spectrum Protect data storage architecture and its implications need to be considered.

How to identify backup and migration candidates

When using IBM Spectrum Protect for backing up IBM Spectrum Scale file systems, there are several choices for the method used to identify backup and migration candidates.

File system and fileset backups with IBM Spectrum Protect

mmbackup supports backup from independent filesets in addition to backup of the whole file system. Independent filesets have similar capabilities as a file system in terms of quota management, dependent filesets, snapshots, and having their own inode space.

Considerations for using fileset backup with IBM Spectrum Protect

IBM Spectrum Protect stores backup data indexed by file or directory object path names in its database. However, file system objects are identified for IBM Spectrum Protect by their object ID extended attributes. In both cases, IBM Spectrum Protect is not aware of the fileset entity. Therefore, the fileset configuration of an IBM Spectrum Scale cluster is neither backed up nor can it be restored by IBM Spectrum Protect alone, unless separate action is taken.

Considerations for backing up file systems that are managed with IBM Spectrum Protect for Space Management

When IBM Spectrum Protect for Space Management is used to migrate data to secondary storage pools (for example, tape volumes), this migrated data might need to be recalled to disk if the file becomes a backup candidate due to certain changes a user may apply to it.

File system and fileset backups with IBM Spectrum Protect

mmbackup supports backup from independent filesets in addition to backup of the whole file system. Independent filesets have similar capabilities as a file system in terms of quota management, dependent filesets, snapshots, and having their own inode space.

Fileset **mmbackup** provides a finer granularity for backup purposes and permits the administrator to:

- Have different backup schedules for different filesets. For example, providing the flexibility that filesets containing more important data to be backed up more often than other filesets.
- Have a fileset dedicated for temporary or transient files or files that do not need to be backed up.
- Use file system backups in conjunction with fileset backup to implement a dual data protection scheme such that file system backup goes to one IBM Spectrum Protect server while fileset backup goes to a different IBM Spectrum Protect server.

Backups of file systems and independent filesets are controlled by using the **mmbackup** option `--scope`.

In the examples used in this topic, the following environment is assumed:

```
tsm server name           => tsm1,tsm2,...
file system device name   => gpfs0 or /dev/gpfs0
file system mountpoint    => /gpfs0

fileset names/junction path
-----
depfset1,depfset2,...     /gpfs0/depfset1,/gpfs0/depfset2           for dependent filesets
indepfset1,indepfset2,... /gpfs0/indepfset1,/ibm/gpfs0/indepfset2 for independent filesets
```

File system backup

File system backup protects the whole file system. The default value for option `--scope` is `filesystem` and thus it can be omitted.

To back up the `gpfs0` file system in its entirety to the IBM Spectrum Protect server named `tsm1`, use the following command:

```
mmbackup gpfs0 --tsm-servers tsm1
mmbackup gpfs0 --tsm-servers tsm1 --scope filesystem
mmbackup /dev/gpfs0 --tsm-servers tsm1
mmbackup /dev/gpfs0 --tsm-servers tsm1 --scope filesystem
mmbackup /gpfs0 --tsm-servers tsm1
mmbackup /gpfs0 --tsm-servers tsm1 --scope filesystem
```

This example shows the file system backup of `gpfs0` using the short or long device name (first 4 lines) or the mount point as a directory input (last 2 lines).

Independent fileset backup

Independent fileset backup protects all files that belong to the inode space of a single independent fileset. The backup of an independent fileset might include other dependent filesets and folders, but not nested independent filesets because each independent fileset has its own inode space.

The following information describes the mmbbackup capability to support independent fileset. The examples with IBM Spectrum Protect show the potential/likely use of this capability.

To back up the independent fileset `indepfset1` of the file system `gpfs0` to IBM Spectrum Protect server named `tsm1`, use the following command:

```
mmbbackup /gpfs0/indepfset1 --tsm-servers tsm1 --scope inodespace
```

Fileset backup is an additional option to existing IBM Spectrum Scale backup or data protection options. Therefore, starting with a file system in which the whole file system has already been backed up on a regular basis, administrators can also start protecting data with fileset backup on some or all filesets. Administrators may choose to utilize fileset backup at any time, with no limit. It is not required to back up the whole file system in either one mode or another unless the chosen data protection approach requires it. Also, administrators can choose different IBM Spectrum Protect servers for some or all fileset backups if it is desired to have these on separate servers. For example, a valid configuration may have one IBM Spectrum Protect server that only gets whole file system backups, and a different one for fileset backups.

Note: When mixing file system and fileset backup on the same IBM Spectrum Protect server, consider that a target file that is handled by both backup approaches gets backed up twice. Each backup activity consumes one storage version of the file. Thus, the same file version is stored twice on the IBM Spectrum Protect server.

If migrating to use only fileset backup to protect the whole file systems, ensure that all independent filesets are backed up and keep the backup of all filesets current. Remember to include a backup of the root fileset as well by using a command such as:

```
mmbbackup /gpfs/gpfs0 --scope inode-space --tsm-servers tsm1
```

To verify the completeness of the data protection using fileset backup only, use the following steps:

1. Identify all independent filesets available in the file system by using the following command:

```
mmllsfileset device -L
```

Identify the independent filesets by the InodeSpace column. The value identifies the corresponding inode space or independent fileset while the first occurrence refers to the corresponding fileset name and fileset path.

Note: Ensure that a backup is maintained of the fileset called `root` that is created when the file system is created and that can be seen as the first fileset of the file system.

2. For every identified independent fileset, verify that a shadow database exists as follows:
 - a. Check for the file `.mmbbackupShadow.<digit>.<TSMserverName>.fileset` in the fileset junction directory.
 - b. If the file exists, determine the time of last backup by looking at the header line of this file. The backup date is stored as last value of this line.

For example:

```
head -n1 /gpfs0/indepfset1/.mmbbackupShadow.*.fileset
%%mshadow0%%:00_BACKUP_FILES_41:1400:/gpfs0:mmbbackup:1:Mon Apr 20 13:48:45 2015
```

Where *Mon Apr 20 13:48:45 2015* in this example is the time of last backup taken for this fileset.

3. If any independent filesets are missing their corresponding `.mmbbackupShadow.*` files, or if they exist but are older than the data protection limit for their backup time, then start or schedule a backup of these filesets.

Note: This action needs to be done for every file system for which the fileset backup approach is chosen.

Note: Backup of a nested independent fileset is not supported. To work around this limitation, unlink the nested fileset and link it at another location in the root fileset prior to beginning to use fileset backup pervasively.

Related concepts

Considerations for provisioning IBM Spectrum Protect servers to handle backup of IBM Spectrum Scale file systems

When backing up IBM Spectrum Scale file systems, the IBM Spectrum Protect server must be configured with adequate resources to handle a much larger load of client data, sessions, and mount points than typically encountered with single-user file systems such as workstations.

IBM Spectrum Protect data storage model

When using IBM Spectrum Protect for backing up IBM Spectrum Scale file systems, the IBM Spectrum Protect data storage architecture and its implications need to be considered.

How to identify backup and migration candidates

When using IBM Spectrum Protect for backing up IBM Spectrum Scale file systems, there are several choices for the method used to identify backup and migration candidates.

Comparison of snapshot based backups and backups from live system

Backing up large file systems can take many hours or even days. When using the IBM Spectrum Scale command **mmbackup**, time is needed for the following steps.

Considerations for using fileset backup with IBM Spectrum Protect

IBM Spectrum Protect stores backup data indexed by file or directory object path names in its database. However, file system objects are identified for IBM Spectrum Protect by their object ID extended attributes. In both cases, IBM Spectrum Protect is not aware of the fileset entity. Therefore, the fileset configuration of an IBM Spectrum Scale cluster is neither backed up nor can it be restored by IBM Spectrum Protect alone, unless separate action is taken.

Considerations for backing up file systems that are managed with IBM Spectrum Protect for Space Management

When IBM Spectrum Protect for Space Management is used to migrate data to secondary storage pools (for example, tape volumes), this migrated data might need to be recalled to disk if the file becomes a backup candidate due to certain changes a user may apply to it.

Considerations for using fileset backup with IBM Spectrum Protect

IBM Spectrum Protect stores backup data indexed by file or directory object path names in its database. However, file system objects are identified for IBM Spectrum Protect by their object ID extended attributes. In both cases, IBM Spectrum Protect is not aware of the fileset entity. Therefore, the fileset configuration of an IBM Spectrum Scale cluster is neither backed up nor can it be restored by IBM Spectrum Protect alone, unless separate action is taken.

IBM Spectrum Scale does provide commands such as **mmbackupconfig** and **mmrestoreconfig** to backup and restore file system configuration data.

Even though backups of independent filesets may be stored on a plurality of IBM Spectrum Protect servers, IBM Spectrum Protect for Space Management can only utilize one server per managed file system if **mmbackup** is used on that file system and sends the backup data to that same server. IBM Spectrum Protect for Space Management multiserver feature is not integrated with the **mmbackup** command.

If space management is enabled for an IBM Spectrum Scale file system, and a plurality of IBM Spectrum Protect servers are utilized for backup, the following limitations apply to the use of IBM Spectrum Protect for Space Management:

- The setting of `migrequiresbkup` cannot be utilized since the server for space management might not have access to the database for the server doing backup on a particular object.
- The *inline copy* feature that allows the backup of a migrated object through copying data internally from the IBM Spectrum Protect for Space Management pool to the backup pool inside the IBM Spectrum Protect server cannot be utilized since migrated data may reside on a different server than the one performing the backup.

The following special considerations apply when using fileset backup with IBM Spectrum Protect:

- IBM Spectrum Protect Backup-Archive client does not restore fileset configuration information. Therefore, a full file system backup is not sufficient for disaster recovery when the whole file system structure needs to be recovered.
- Nested independent filesets cannot be backed up when using the fileset backups. Resolve the nesting before starting to protect the data with **mmbackup** on each fileset.
- The following operations are discouraged: fileset unlink, fileset junction path change via unlink and link operations, fileset deletion. These operations will incur a heavy load on the IBM Spectrum Protect server at the next backup activity due to the significant alteration of the path name space that result from these operations. If a fileset must be moved in the file system, be prepared for the extra time that might be required to run the next backup.
- If only utilizing fileset backup, be sure to include every fileset, including the root fileset to ensure complete data protection.
- When both file system and fileset backups use the same IBM Spectrum Protect server, new and changed files may be unintentionally backed up twice. This consumes two of the file versions stored in the IBM Spectrum Protect server with the same file data.

Loss of data protection may occur if a fileset is unlinked or deleted for an extended period of time and the objects formerly contained in that fileset are allowed to expire from the backup sets.

- Do not move, rename, or relink a fileset, unless a new full backup of the fileset is expected to be made immediately after the restructuring occurs. IBM Spectrum Protect server only recognizes objects in backup only as long as they remain in their original path name from the root of the file tree (mount point).

The following special considerations apply when using fileset backup and IBM Spectrum Protect for Space Management:

- When using fileset backup to different IBM Spectrum Protect servers, the server setting **migrequiresbkup** must be set to NO to be able to migrate files.
- Automatic server selection for backup by **mmbackup** is not supported and therefore the IBM Spectrum Protect *multiple space management server* feature is not integrated with **mmbackup** data protection of IBM Spectrum Scale. Using fileset backup to different IBM Spectrum Protect servers in conjunction with the *multiple space management server* feature requires special handling and configuration because backup and space management rules must match so that files are handled by the same IBM Spectrum Protect server.

Related concepts

[Considerations for provisioning IBM Spectrum Protect servers to handle backup of IBM Spectrum Scale file systems](#)

When backing up IBM Spectrum Scale file systems, the IBM Spectrum Protect server must be configured with adequate resources to handle a much larger load of client data, sessions, and mount points than typically encountered with single-user file systems such as workstations.

[IBM Spectrum Protect data storage model](#)

When using IBM Spectrum Protect for backing up IBM Spectrum Scale file systems, the IBM Spectrum Protect data storage architecture and its implications need to be considered.

[How to identify backup and migration candidates](#)

When using IBM Spectrum Protect for backing up IBM Spectrum Scale file systems, there are several choices for the method used to identify backup and migration candidates.

[Comparison of snapshot based backups and backups from live system](#)

Backing up large file systems can take many hours or even days. When using the IBM Spectrum Scale command **mmbackup**, time is needed for the following steps.

[File system and fileset backups with IBM Spectrum Protect](#)

mmbackup supports backup from independent filesets in addition to backup of the whole file system. Independent filesets have similar capabilities as a file system in terms of quota management, dependent filesets, snapshots, and having their own inode space.

Considerations for backing up file systems that are managed with IBM Spectrum Protect for Space Management

When IBM Spectrum Protect for Space Management is used to migrate data to secondary storage pools (for example, tape volumes), this migrated data might need to be recalled to disk if the file becomes a backup candidate due to certain changes a user may apply to it.

Considerations for backing up file systems that are managed with IBM Spectrum Protect for Space Management

When IBM Spectrum Protect for Space Management is used to migrate data to secondary storage pools (for example, tape volumes), this migrated data might need to be recalled to disk if the file becomes a backup candidate due to certain changes a user may apply to it.

Recall will automatically occur synchronously if the user attempts to change the file data. However, no synchronous recall occurs in the following scenarios:

- If a user changes the owner, group, mode, access control list, or extended attributes of the file.
- If the user renames the file or any of the parent directories of the file.

Any of these changes, however, does make the file a candidate for the next **mmbackup** invocation. When **mmbackup** supplies the path to a migrated file to IBM Spectrum Protect for backup, synchronous recall will occur. This can lead to a "Recall Storm" in which tape library and IBM Spectrum Protect server resources become overburdened handling many simultaneous recall requests. Disk space is also consumed immediately by recalled data. These are undesirable outcomes and can interfere with the normal processing of the customer's workload.

Since **mmbackup** is able to discern that the data was migrated for such files, it will defer the backup, and instead append a record referring to the file's path name into a special file in the root of the file system, or fileset in the case of fileset-level backup. This list can then be used by the system administrator to schedule recall of the deferred files data to permit the backup to occur on the next invocation of the **mmbackup** command. By deferring the recall to the system administrator, **mmbackup** prevents unexpected, excessive tape library activity which might occur if many such migrated files were nominated for backup due to user actions such as renaming a high level directory or changing owner, group, or modes on many files.

The system administrator must schedule the recall for the migrated objects omitted by **mmbackup** according to the availability of tape and disk resources in the cluster. For information about optimized tape recall, see *Optimized tape recall processing* in IBM Spectrum Protect for Space Management documentation.

Related concepts

Considerations for provisioning IBM Spectrum Protect servers to handle backup of IBM Spectrum Scale file systems

When backing up IBM Spectrum Scale file systems, the IBM Spectrum Protect server must be configured with adequate resources to handle a much larger load of client data, sessions, and mount points than typically encountered with single-user file systems such as workstations.

IBM Spectrum Protect data storage model

When using IBM Spectrum Protect for backing up IBM Spectrum Scale file systems, the IBM Spectrum Protect data storage architecture and its implications need to be considered.

How to identify backup and migration candidates

When using IBM Spectrum Protect for backing up IBM Spectrum Scale file systems, there are several choices for the method used to identify backup and migration candidates.

Comparison of snapshot based backups and backups from live system

Backing up large file systems can take many hours or even days. When using the IBM Spectrum Scale command **mmbackup**, time is needed for the following steps.

File system and fileset backups with IBM Spectrum Protect

mmbackup supports backup from independent filesets in addition to backup of the whole file system. Independent filesets have similar capabilities as a file system in terms of quota management, dependent filesets, snapshots, and having their own inode space.

Considerations for using fileset backup with IBM Spectrum Protect

IBM Spectrum Protect stores backup data indexed by file or directory object path names in its database. However, file system objects are identified for IBM Spectrum Protect by their object ID extended attributes. In both cases, IBM Spectrum Protect is not aware of the fileset entity. Therefore, the fileset configuration of an IBM Spectrum Scale cluster is neither backed up nor can it be restored by IBM Spectrum Protect alone, unless separate action is taken.

Planning for Quality of Service for I/O operations (QoS)

With the Quality of Service for I/O operations (QoS) capability, you can run a GPFS maintenance command without the risk of it dominating file system I/O performance and significantly delaying other tasks.

The GPFS *maintenance commands* are a group of about 20 potentially long-running, I/O-intensive commands that can generate hundreds or thousands of requests for I/O operations per second. This high demand for I/O can greatly slow down normal tasks that are competing for the same I/O resources. With QoS, you can assign an instance of a maintenance command to a QoS class that has a lesser I/O priority. Although the instance of the maintenance command now takes longer to run to completion, normal tasks have greater access to I/O resources and run more quickly. The maintenance commands are listed in the help topic for the *mmchqos* command in the *IBM Spectrum Scale: Command and Programming Reference*.

Note the following requirements and limitations:

- QoS requires the file system to be at V4.2.0.0 or later.
- 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 I/O operations per second (IOPS) but does not regulate them.
- When you change allocations, mount the file system, or re-enable QoS, a brief delay due to reconfiguration occurs before QoS starts applying allocations.

To set up QoS, you allocate shares of I/O operations per second (IOPS) to two QoS classes, *maintenance* and *other*. Each storage pool has these two classes. The *maintenance* class is the class in which the GPFS maintenance commands run by default. Typically you assign a smaller share of IOPS to this class so that the commands in it do not overwhelm the file system with I/O requests. The *other* class is the class to which all other processes belong. Typically you assign an unlimited number of IOPS to this class so that normal processes have greater access to I/O resources and finish more quickly.

When a process issues an I/O request, QoS first finds the QoS class to which the process belongs. It then finds whether the class has any I/O operations available for consumption. If so, QoS allows the I/O operation. If not, then QoS queues the request until more I/O operations become available for the class. For a particular storage pool, all instances of maintenance commands that are running concurrently compete for the IOPS that you allocated to the *maintenance* class of that storage pool. Similarly, all processes that belong to the *other* class compete for the IOPS that you allocated to the *other* class of that storage pool. If no processes in the *maintenance* class are running, processes in the *other* class can also compete for the *maintenance* IOPS.

If you do not configure a storage pool for QoS, then processes compete for I/O operations in that storage pool in the normal way without interference.

By default, for both the *maintenance* class and the *other* class, QoS divides the IOPS allocation equally among the nodes of the cluster that have the file system mounted. For example, you might assign 1000 IOPS to the *maintenance* class of storage pool SP1. If your allocation applies to all the nodes in the cluster (the default setting) and the cluster has five nodes, then the *maintenance* class of each node gets an allocation of 200 IOPS for storage pool SP1. At run time, on each node all processes that belong to the *maintenance* class and that access storage pool SP1 compete for the 200 IOPS.

You can also divide IOPS among a list of nodes, a user-defined node class, or the nodes of a remote cluster that is served by the file system.

To get started with QoS, follow the steps that are outlined in the topic *Setting the Quality of Service for I/O operations (QoS)* in *IBM Spectrum Scale: Administration Guide*.

Use the `mmchqos` command to allocate IOPS to the QoS classes `maintenance` and `other`. You can set up allocations for one pool or several pools in one call to the command. You can designate a single node, several nodes, or all the nodes in the current cluster to participate in the QoS operations. You can also designate the nodes in a remote cluster to participate. To preserve a particular configuration, you can set up a stanza file that the `mmchqos` command reads when it runs.

You can reset or change IOPS allocations at any time without unmounting the file system. You can also disable QoS at any time without losing your IOPS allocations. When you reenables QoS, it resumes applying the allocations.

Remember the following points:

- You can allocate shares of IOPS separately for each storage pool.
- QoS divides each IOPS allocation equally among the specified nodes that mounted the file system.
- Allocations persist across unmounting and remounting the file system.
- QoS stops applying allocations when you unmount the file system and resumes when you remount it.

Use the `mm1sqos` command to display the consumption of IOPS. The command displays data for subperiods within the period that you specify. You can configure the type of information that is displayed. You can display information for specified storage pools or for all storage pools. You can display information for various time periods, from 1 second up to about 15 minutes. You can display I/O performance for each QoS class separately or summed together. You can also display the I/O performance of each node separately or summed across the participating nodes.

Planning for extended attributes

IBM Spectrum Scale supports extended attributes for files, directories, block devices, pipes, and named sockets in a file system.

Extended file attributes are key-value pairs that can be set programmatically by the file system, by other middleware such as the Data Management API, by the operating system, or by users. The name of an extended attribute consists of a namespace name followed by a dot followed by an attribute name, as in the following extended attribute names:

```
gpfs.Encryption
user.swift.metadata
system.posix_acl_access
```

The public namespaces are *system*, *security*, *trusted*, and *user*. IBM Spectrum Scale has a few private namespaces, which include *gpfs* and *dmapi*.

You can access extended attributes in the public namespaces through operating system commands and functions. Linux has commands such as `setfattr` and `getfattr` and functions such as `setxattr()` and `getxattr()` for accessing extended attributes.

- In Linux, the following facts are relevant:
 - You must have root-level access to work with extended attributes in the *system*, *security*, and *trusted* namespaces.
 - To set an extended attribute for a file or file object (directory, block device, pipe, or named socket) you must have write access to the entity.
 - The `cp --preserve=xattr` Linux command copies either the POSIX or the NFSv4 ACL extended attributes when an IBM Spectrum Scale file is copied. Also some Linux system calls are extended to handle attributes for POSIX and NFSV4 ACLs when the system calls are applied to files in IBM Spectrum Scale file systems. For more information, see *Managing GPFS access control lists* in *IBM Spectrum Scale: Administration Guide*.

- In Windows, the following facts are relevant:
 - You must log on as an administrator to work with extended attributes in the *system*, *security*, and *trusted* namespaces.
 - Windows has file permissions for reading and writing extended attributes that are separate from its file permissions for reading or writing file data.
- In AIX, the following facts are relevant:
 - IBM Spectrum Scale does not support AIX commands or functions that access extended attributes.

The native operating system commands and functions that are referred to in the preceding paragraph cannot access the private namespaces of IBM Spectrum Scale. You can access the extended attributes in the IBM Spectrum Scale private namespaces only through IBM Spectrum Scale resources, which are described in the following list. These resources can also access the extended attributes in the public namespaces. IBM Spectrum Scale supports these resources on all platforms, Linux, Windows, and AIX:

- The rules for automating file management include a set of functions for accessing, setting, and evaluating extended attributes. For more information, see *Extended attribute functions* in the *IBM Spectrum Scale: Administration Guide*.
- The **mmchattr** and **mmlsattr** commands provide parameters to create, set, get, or delete extended attributes. For more information, see *mmchattr command* and *mmlsattr command* in the *IBM Spectrum Scale: Command and Programming Reference* guide.

Note: The **mmchattr** command does not set some DMAPI extended attributes because of their side effects. However, you can set these extended attributes through DMAPI. For more information, see *IBM Spectrum Scale Data Management API for GPFS information* in the *IBM Spectrum Scale: Command and Programming Reference* guide.

- The GPFS programming interfaces include a set of subroutines for getting and setting extended attributes. For more information, see *GPFS programming interfaces* in the *IBM Spectrum Scale: Command and Programming Reference* guide.

Fast extended attributes are extended attributes that are stored in the inode of the file or file object so that they can be read or written quickly. Some features of IBM Spectrum Scale require support for fast extended attributes, such as DMAPI, file encryption, and file heat. To work with fast extended attributes, you use the same methods that you use with regular extended attributes.

The following features in IBM Spectrum Scale also handle extended attributes:

- Limitations:
 - Extended attributes are not encrypted when a file is encrypted.
 - Extended attributes cannot be added, deleted, or modified in an immutable or an appendOnly file. This restriction does not apply to extended attributes in the **dmapi** or **gpfs** namespaces.
- Size:
 - To determine the maximum allowed size of extended attributes, see Q6.11 in [IBM Spectrum Scale FAQ in IBM Documentation](#).
 - Fast extended attributes are stored first in the inode of a file, then, if more space is needed, in a maximum of one overflow block. The overflow block is the same size as the current metadata block size. The maximum space in the overflow block for storing extended attributes is the minimum of 64 KB or the metadata block size.
- Administration commands:
 - The **mmafmconfig** command has parameters that enable or disable extended attributes or sparse file support in the AFM cache.
 - The **mmclone** command does not copy extended attributes.
 - See the administration commands in the item "Backup and restore" that follows.
- Encryption:

- In encryption, file extension keys (FEKs) are stored in the `gpfs` .Encryption extended attribute. It is a good idea to create a file system with an inode size of 4 K or larger to accommodate FEKs that are encrypted several times. If you encounter an error message that says that the extended attributes are too large, you must either change the encryption policy so that the file key is wrapped fewer times, reduce the number of keys that are used to wrap a file key, or create a file system that has a larger inode size. For more information, see the following links:
 - *Encryption policies in the IBM Spectrum Scale: Administration Guide*
 - *Preparation for encryption in the IBM Spectrum Scale: Administration Guide*
 - Error message 6027-3470 [E] in the *IBM Spectrum Scale: Problem Determination Guide*
- Backup and restore:
 - Scale Out Backup and Restore (SOBAR) preserves and restores extended attributes. For more information, see *Scale Out Backup and Restore (SOBAR) in the IBM Spectrum Scale: Administration Guide*.
 - The **mmbackup** and **mmrestorefs** commands preserve extended attributes. The **mmrestorefs** command has options to restore encryption attributes and to not restore external attributes. The IBM Spectrum Protect product has a setting in the `dsm.sys` configuration file that omits backing up a file if only an extended attribute is changed (SKIPACLUPDATECHECK). For more information, see *Options in the IBM Spectrum Protect configuration file dsm.opt* in the *IBM Spectrum Scale: Administration Guide*.
 - The **mmrestorefs** command has options to restore encryption attributes and to not restore external attributes.
- Disaster recovery
 - In outband disaster recovery, if you are transferring files that contain extended attributes, such as object data, you must use a method that transfers extended attributes, such as IBM Spectrum Protect, cross-cluster mount, or AFM.
- Objects
 - In unified file and object access, the identity management modes **local_mode** and **unified_mode** have settings for retaining extended attributes. You can set values in the `object-server-sof.conf` configuration file to copy extended attributes, Windows attributes, and ACLs from an existing object. For more information, see the following links:
 - *Identity management modes for unified file and object access in the IBM Spectrum Scale: Administration Guide*
 - *Configuring authentication and ID mapping for file access in the IBM Spectrum Scale: Administration Guide*
- NFS v4
 - NFS v4 uses extended attributes to store some of its file attributes. For more information, see *Linux ACLs and extended attributes* in the *IBM Spectrum Scale: Administration Guide*.

Fast extended attribute support is now the default on newly created file systems. You can verify that a file system had fast extended attributes enabled by issuing the following command:

```
mmfsfs <Device> --fastea
```

Where, `<Device>` is the name of the file system, for example, `gpfs1`. If the command output shows that fast extended attributes are not enabled, then run the following command to enable fast extended attributes:

```
mmigratefs <Device> --fastea
```

Where, `<Device>` is the name of the file system. For more information, see [“Completing the upgrade to a new level of IBM Spectrum Scale” on page 572](#).

Planning for the Highly Available Write Cache feature (HAWC)

Learn about the Highly Available Write Cache feature (HAWC).

Components that HAWC interacts with

HAWC interacts with several fundamental components of IBM Spectrum Scale. You might want to review these components before you read about HAWC.

System storage pool

The *system storage pool* or *system pool* is a required storage pool that contains information that IBM Spectrum Scale uses to manage a file system. Each file system has only one system storage pool, which is automatically created when the file system is created. The system storage pool contains the following types of information:

- Control information (such as file system control structures, reserved files, directories, symbolic links, special devices)
- The metadata associated with regular files, including indirect blocks and extended attributes
- Regular file data, if the `usage=dataAndMetadata` option is set in the NSD stanza for a system storage pool NSD
- The file system recovery logs (default location)

System.log storage 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 Spectrum 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).

File system recovery logs

A *file system recovery log* is a write-ahead log or journal of I/O metadata that describes pending write operations for a node of a file system. In IBM Spectrum Scale, it is also sometimes referred to as the recovery log, the GPFS log, or the IBM Spectrum Scale log. IBM Spectrum Scale creates and maintains a separate recovery log for every node that mounts a file system. Recovery logs are stored in the system storage pool by default or in the *system.log* storage pool if one is defined. The recovery logs can be read by any node that mounts the file system. If a node is unexpectedly shut down while write operations are pending for one of its hard disks, IBM Spectrum Scale can read the recovery log for the failed node and restore the file system to a consistent state. The recovery can occur immediately, without having to wait for the failed node to return.

The recovery logs are also used by HAWC to temporarily store HAWC write data and metadata.

Page pool

The *page pool* is an area of pinned memory (memory that is never paged to disk) that contains file data and metadata associated with in-progress I/O operations. When IBM Spectrum Scale processes a file write operation, the first step is putting the write data and metadata for the write operation into the page pool. At an appropriate time, another thread writes the data to a hard disk and removes it from the page pool.

HAWC operation

The high-availability write cache is a disk-caching component that includes caching software and nonvolatile storage. HAWC also uses the file system recovery logs, in which the file system records metadata about its pending write operations. For HAWC purposes, the recovery logs must be located in nonvolatile storage.

When a file write operation arrives at a node, the first part of the processing is the same whether HAWC is active or not. The write data and metadata are copied into an entry in the page pool and the entry is added to a list of similar entries that are waiting for processing. When the entry is processed, the processing depends on whether HAWC is active.

Note: If the write operation is nonsynchronous, it returns to its caller after its write data and metadata are copied into the page pool entry. If the write operation is synchronous, it waits for a notification that the file data has been written to disk.

When HAWC is not active, the write data is copied from the page pool entry and written to the file on hard disk. If the write operation is synchronous, the system notifies the write operation that the write is successful and it returns to its caller.

When HAWC is active, the write data can take either of two paths:

- If the write operation is synchronous and the size of the file data is less than or equal to the write data threshold, HAWC copies the file data from the page pool entry into the recovery log, along with any I/O metadata that is required for recovery. The *write data threshold* variable is set by the **mmcrfs** command or the **mmchfs** command. Next HAWC notifies the original write operation that the file data is successfully written to hard disk. In fact, the file data is not written to hard disk yet, although it is preserved in the recovery log as a backup. HAWC then starts a write-behind thread that eventually writes the file data to the hard disk. When the data is safely written, HAWC purges the file data and I/O metadata from the recovery log, because it is no longer needed.
- If the write operation is not synchronous or if the size of the write data is greater than the write cache threshold, then the write data follows the same path that is followed when HAWC is not active. The system copies the write data from the page pool entry and writes it to hard disk. If the original write operation is synchronous, the system notifies it that the file data is safely written to the hard disk.

HAWC improves the performance of small synchronous write operations in two ways. First, it allows synchronous write operations to return the calling application as soon as the write data is written into the recovery log. The calling application does not have to wait for the much lengthier process of writing the data to hard disk. Second, the HAWC caching software can consolidate small sequential writes into one larger write. This consolidation eliminates all but one of the initial disk seeks that is required if the data is written as multiple writes.

The write-cache threshold variable can be adjusted by specifying a value for the **--write-cache-threshold** parameter of the **mmchfs** command. The valid range is 0 - 64 K in multiples of 4 K. You can also set this variable when you create the file system by specifying the same parameter in the **mmcrfs** command. Setting the write cache threshold to zero disables HAWC. You can update the write threshold variable at any time; the file system does not have to be mounted on the node.

HAWC storage scenarios

You can set up the HAWC storage in either of two configurations or scenarios. In the first scenario, the nonvolatile storage is located in a centralized fast storage device, such as a controller with SSDs:

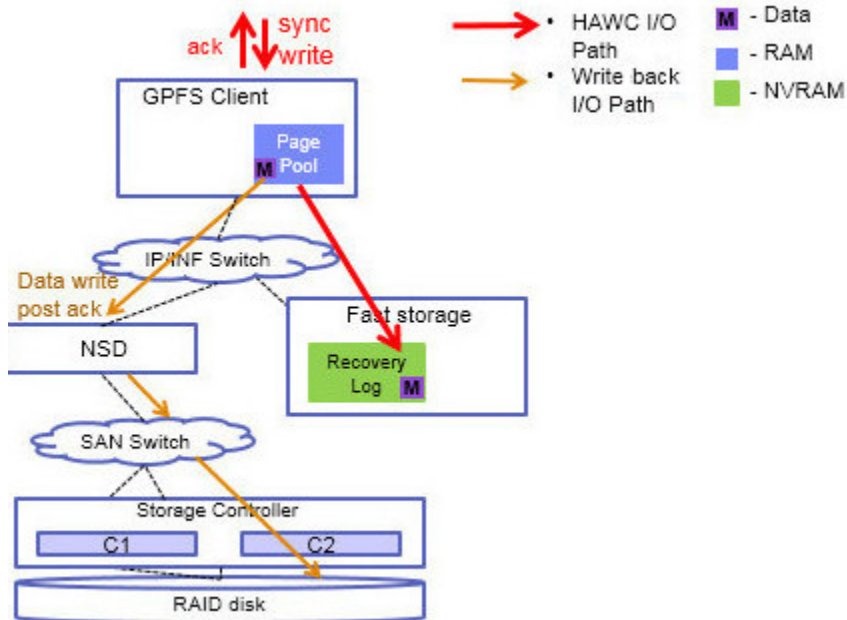


Figure 33. Shared fast storage

In this scenario, when a synchronous write operation arrives at a node, the file data and metadata are copied a page pool entry in the usual way. If the size of the file data is less than the write data threshold, HAWC copies the file data into the recovery log along with any I/O metadata that is required for recovery. Next, HAWC returns an acknowledgment to the write operation that indicates that the file data is successfully written to hard disk. HAWC then starts a write-behind thread that eventually writes the file data to the hard disk. When the data is safely written, HAWC purges the file data and I/O metadata for the write operation from the recovery log.

In the second scenario, the nonvolatile storage consists of multiple storage devices that are distributed across some or all of the nodes in the cluster:

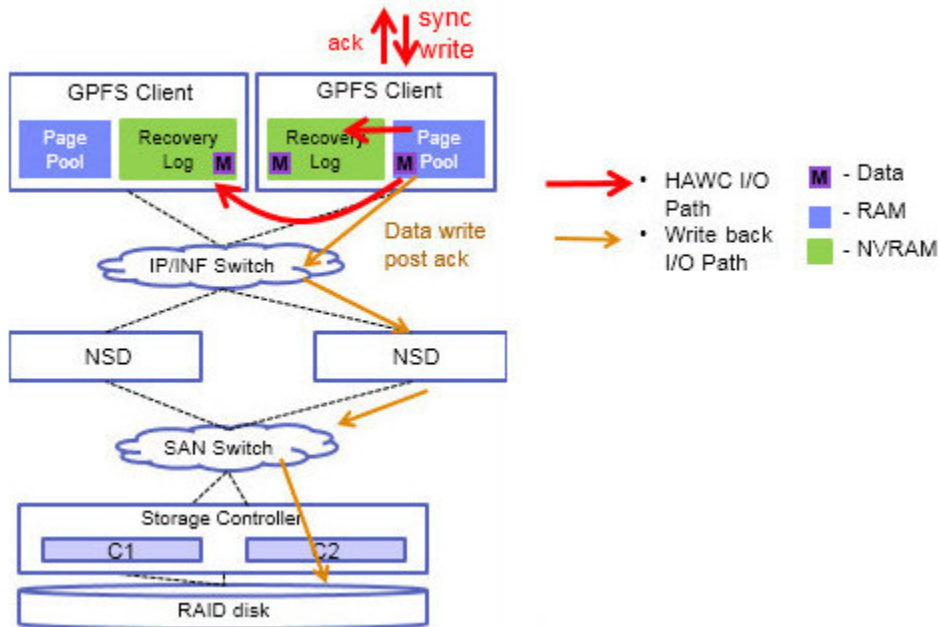


Figure 34. Distributed fast storage

Although the hardware configuration is different in the second scenario, the data flow is similar to the data flow of the first scenario. The synchronous write operation arrives at a node and the write data and

I/O metadata are written into a page pool entry. If the size of the write data is smaller than the write storage threshold, HAWC copies the file data and relevant I/O metadata to the recovery log. The data is striped over the various disks that belong to the recovery log storage pool. HAWC returns a successful acknowledgment to the synchronous write operation and starts a write-behind thread that later writes the file data from the page pool entry to a hard disk. When the data is safely written, HAWC purges the file data and I/O metadata from the recovery log.

HAWC software configuration

After you install the nonvolatile storage devices, complete the following steps to configure IBM Spectrum Scale for HAWC. These steps assume that you are adding HAWC storage to an existing file system:

1. Stop the GPFS daemon on all the nodes of the cluster.
2. Create NSD stanzas for the nonvolatile storage devices. In the stanza, specify one storage pool for all the nonvolatile storage devices, which must be either the system pool or the system.log pool.
3. Run **mmcrnsd** to create the NSDs.
4. Run **mmaddisk** to add the NSDs to the file system and to create the system.log pool if necessary.
5. Start the GPFS daemons on all nodes.
6. Optionally, run the **mmchfs** command with the **-L** parameter to set the size of the recovery logs to a non-default value.
7. Optionally, run the **mmchfs** command with the **--log-replicas** parameter to set the number of replicas of the recovery log to a non-default value. This option is applicable only if the recovery logs are stored in the system.log pool.
8. To activate HAWC, run the **mmchfs** command with the **--write-cache-threshold** parameter set to a nonzero value.

HAWC is now active.

Planning for systemd

IBM Spectrum Scale supports systemd version 219 and later on Linux operating systems.

IBM Spectrum Scale automatically installs and configures itself as a systemd service in systems that have systemd version 219 or later installed. The service unit files are installed in the systemd `lib` directory, as shown in the following list:

- Red Hat Enterprise Linux and SUSE Linux Enterprise Server:

```
/usr/lib/systemd/system
```

- Ubuntu:

```
/lib/systemd/system
```

IBM Spectrum Scale includes the following systemd services. For references to IBM Spectrum Scale daemons, see the descriptions that immediately follow this list:

gpfs

Starts or stops the GPFS daemon (**mmfsd**).

gpfs-online

Marker service which switches to active after GPFS has reached a working state. It uses the GPFS state as shown in the **mmhealth node show** command output for its status.

gpfs-wait-mount.service

Marker service which switches to active after the GPFS filesystems have been mounted. It uses the filesystem state as shown in the **mmhealth node show** command output for its status. The status for the **mmhealth** command only looks at filesystems which have automount value set to yes.

mmccrmonitor

Starts or stops the **mmccrmonitor** daemon. This service is enabled to start at boot time. It runs all the time. If it is killed by some means other than the systemd **systemctl** command, restart it manually with the **systemctl** command.

mmautoload

Starts the **mmautoload** daemon after the node is rebooted and also starts the GPFS daemon if autostart is specified. This service also shuts down the GPFS daemon when the operating system shuts down.

The systemd manager invokes this service when the node is shut down or rebooted. This service is enabled to start at boot time after **mmccrmonitor** is started. It cannot be started or stopped with the systemd **systemctl** command.

mmsdrserv

Starts or stops the **mmsdrserv** daemon. This service is started after the node is rebooted and is stopped after the GPFS daemon is started. It is also stopped as needed by other GPFS systemd services.

mmsysmon

Starts or stops system health monitoring in a cluster in which the Clustered Configuration Repository (CCR) is enabled. This service is enabled to start when the node is booted and is meant to run continuously until the node is shut down. It also can be started by other services.

lxtrace

Does trace operations. This service is controlled by the **mmtracectl** command. Do not start or stop this service with the **systemctl** command.

Descriptions of the IBM Spectrum Scale daemons:

- The GPFS daemon (**mmfsd**) is the main IBM Spectrum Scale daemon that runs on the node.
- The **mmccrmonitor** daemon starts and stops the **mmsdrserv** daemon if **mmfsd** is not running.
- The **mmsdrserv** daemon provides access to configuration data to the rest of the nodes in the cluster when the **mmfsd** daemon is not running.
- The **mmautoload** daemon sets up GPFS resources when the operating system is started.

Planning for protocols

This topic describes the planning considerations for using protocols.

Some protocol use considerations:

- At time of release, several features in GPFS are not explicitly tested with protocol functionality. Explicitly not tested features are Local Read Only Cache, Multi-cluster, Encryption, File Placement Optimizer, and Hierarchical Storage Management. They are expected to work with protocols and are tested with protocols over time. However, if you use one of these features before IBM claims support of it, ensure that it is tested with the expected workloads before putting into production.

For protocol support on remote clusters, see *Using NFS/SMB protocol over remote cluster mounts* topic in the *IBM Spectrum Scale: Administration Guide*.

- Use of Clustered NFS (CNFS) is not compatible with use of Clustered Export Service (CES). You must choose one or the other. CNFS (which uses kernel NFS, a different NFS stack than the stack used by the CES infrastructure) continues to be supported; however, if you choose CNFS, you cannot take advantage of the integration of SMB and Object server functionality. If you choose to migrate from CNFS to CES, the CES infrastructure does not support a complete equivalent of CNFS group feature to control failover of IP addresses.
- For information regarding specific limitations about protocols and their integration with GPFS, see the [IBM Spectrum Scale FAQ in IBM Documentation](#) and [IBM Spectrum Scale in IBM Documentation](#).

Authentication considerations

To enable read and write access to directories and files for the users on the IBM Spectrum Scale system, you must configure user authentication on the system. Only one user authentication method, and only one instance of that method, can be supported.

The following matrix gives a quick overview of the supported authentication configurations with the IBM Spectrum Scale system for both file and object access.

- ✓: Supported
- X: Not supported
- NA: Not applicable

Table 29. General authentication support matrix								
Authentication method	ID-mapping method	File						Object
		SMB	SMB with Kerberos	NFSV3	NFSV3 with Kerberos	NFSV4	NFSV4 with Kerberos	
User-defined	User-defined	NA	NA	NA	NA	NA	NA	✓
LDAP with TLS	LDAP	✓	NA	✓	NA	✓	NA	✓
LDAP with Kerberos	LDAP	✓	✓	✓	✓	✓	✓	NA
LDAP with Kerberos and TLS	LDAP	✓	✓	✓	✓	✓	✓	NA
LDAP without TLS and without Kerberos	LDAP	✓	NA	✓	NA	✓	NA	✓
LDAP with SSL		NA	NA	NA	NA	NA	NA	✓
AD	Automatic	✓	✓	X	X	X	X	✓
AD	RFC2307	✓	✓	✓	✓	✓	✓	✓
AD	LDAP	✓	✓	✓	X	X	X	✓
AD with SSL		NA	NA	NA	NA	NA	NA	✓
AD with TLS		NA	NA	NA	NA	NA	NA	✓
NIS	NIS	NA	NA	✓	NA	✓	NA	NA
Local	None	NA	NA	NA	NA	NA	NA	✓
Local (OpenStack Keystone)	None	NA	NA	NA	NA	NA	NA	✓
Local (OpenStack Keystone) with SSL	None	NA	NA	NA	NA	NA	NA	✓

Note:

- NIS is not supported for Object protocol.
- When you use a unified file and object access (serving the same data with a file and with an object), select the appropriate authentication service. For more information, see the *Administering unified file and object access* topic in the *IBM Spectrum Scale: Administration Guide*.
- The ID-mapping option that is given in this table is only applicable for file access. Ignore the ID-mapping details that are listed in the table if you are looking for the supported configurations for object access.
- In the User-defined mode, the customer is free to choose the authentication and ID-mapping methods for file and object and manage on their own. That is, the authentication needs to be configured by the administrator outside of the IBM Spectrum Scale commands and ensure that it is common and consistent across the cluster.
- If LDAP-based authentication is used, ACL management for SMB is not supported.

Unified Identity between Object & File: In this case, we need to ensure that the users get the same user UID and GID across NFS, SMB, and Object. Therefore, only the following authentication mechanisms are supported:

- Object that is configured with AD, and a file is configured with the same AD where the user or group ID is available on AD+RFC 2307.
- Object that is configured with LDAP, and a file is configured with the same LDAP where the user or group ID is available on LDAP.

For more information, see the *Administering unified file and object access* topic in the *IBM Spectrum Scale: Administration Guide*.

The following diagram shows the high-level overview of the authentication configuration.

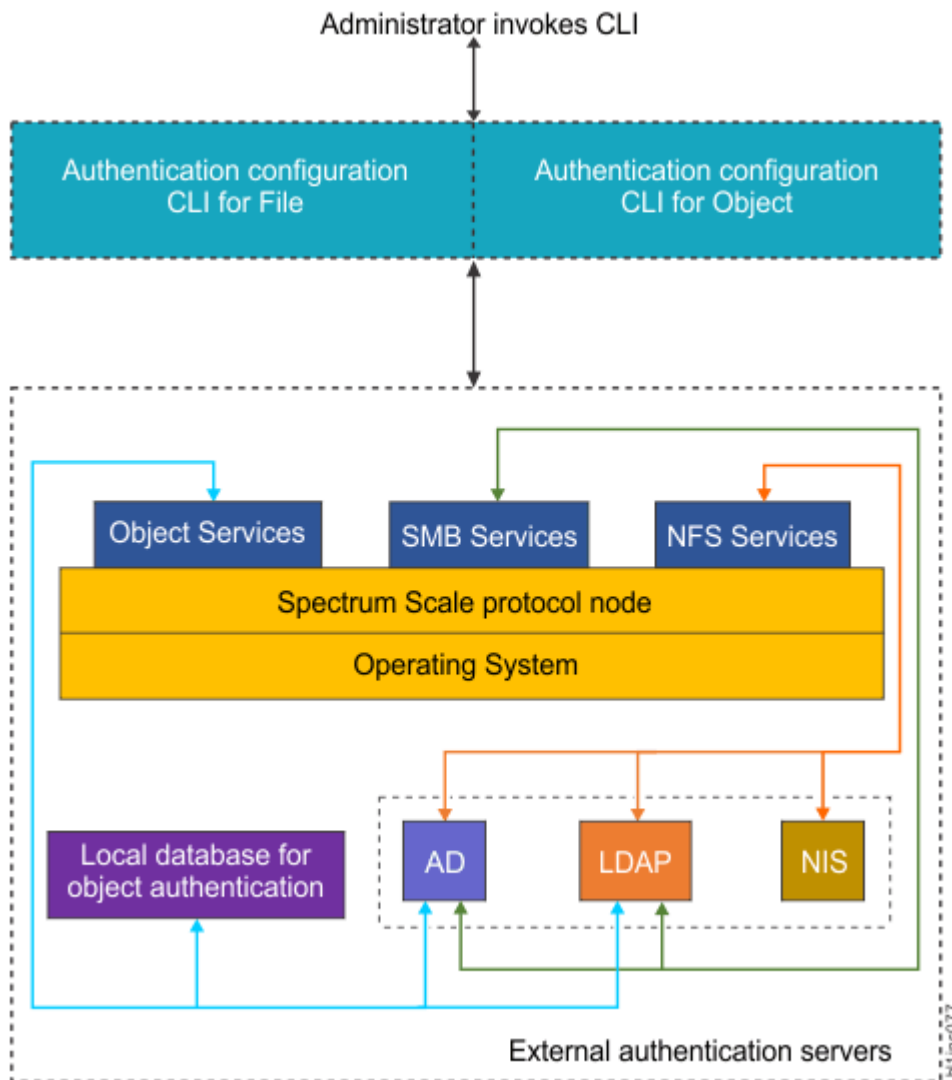


Figure 35. High-level overview of protocol user authentication

The authentication requests that are received from the client systems are handled by the corresponding services in the IBM Spectrum Scale system. For example, if a user needs to access the NFS data, the NFS services resolves the access request by interacting with the corresponding authentication and ID-mapping servers.

For more information about how to configure authentication, see *Managing protocol user authentication* in the *IBM Spectrum Scale: Administration Guide*.

For more planning information, for example, prerequisites, see *Configuring authentication and ID mapping for file access* in the *IBM Spectrum Scale: Administration Guide*.

Impacts of authentication on enabling and disabling protocols

The following are the recommendations for enabling and disabling protocols.

- If authentication is already configured in the system, then the protocols that are enabled in the system to access data cannot be disabled, unless the system administrator cleans up the authentication configuration.
- If authentication is configured, a disabled protocol cannot be enabled. Hence, it is vital to plan properly to decide the protocols that you want to avail now and in near future and enable them correctly in advance.

- If authentication method chosen is active directory (AD), then authentication cannot be configured unless the SMB protocol is enabled and started.
- Protocols cannot be disabled unless authentication is unconfigured. Disabling of a protocol is a critical action and it might result in loss of access to the data hosted by that protocol or other protocols. After you disable the protocol, the authentication can be reconfigured with the same authentication servers and types so that the other enabled protocols can start functioning again.

Note: Instead of disabling protocols, the system administrators might stop a protocol if they do not need it. Stopping a protocol is the recommended action instead of disabling the protocol. Stopping a protocol is as good as removing the protocol from the cluster but it does not have an impact on the access to the data. Users can retain the access when the protocol is started again. Use the **mmces stop -nfs -a** command to stop the NFS protocol on the CES nodes. For more details about these options, see **mmces command** in *IBM Spectrum Scale: Command and Programming Reference*. If a service is stopped and not disabled, then it starts automatically post the system reboot.

- The restrictions for enabling or disabling protocols are not applicable if user-defined authentication method is used for file or object access.
- The authentication configuration needs to be removed to enable a new protocol in the system. If other protocols are already enabled in the system, the system administrator must reconfigure the authentication with the same authentication servers and types. Because of this authentication reconfiguration, the other enabled protocols can start functioning again.

You can also remove ID mappings, along with authentication, if you want to completely remove the authentication configuration. This authentication configuration removal results in permanent loss of access to the data.

Related concepts

Authentication and ID mapping for file access

The system supports an external authentication service to authenticate users on the system. Before you configure an authentication method, ensure that the external authentication service is set up correctly.

Authentication for object access

The OpenStack identity service that is enabled in the system confirms an incoming request by validating a set of credentials that are supplied by a user. The identity management consists of both authentication and authorization processes.

Deleting authentication and ID mapping

You can choose to delete the ID mappings that are generated and stored on IBM Spectrum Scale. The authentication method that is configured in the system can be deleted too. This deletion might result in a complete loss of data access. Before you delete an ID mapping, determine how to maintain data access if needed.

Authentication and ID mapping for file access

The system supports an external authentication service to authenticate users on the system. Before you configure an authentication method, ensure that the external authentication service is set up correctly.

The following steps are involved in user authentication for file access:

1. User tries to connect to the IBM Spectrum Scale system by using their credentials.
2. The IBM Spectrum Scale system contacts the authentication server to validate the user.
3. The IBM Spectrum Scale system contacts the ID map server that provides UIDs and GIDs of the user and user group to verify the identity of the user.
4. If the user credentials are valid, the user gains access to the system.

The following diagram shows the high-level flow of authentication for File-based protocols.

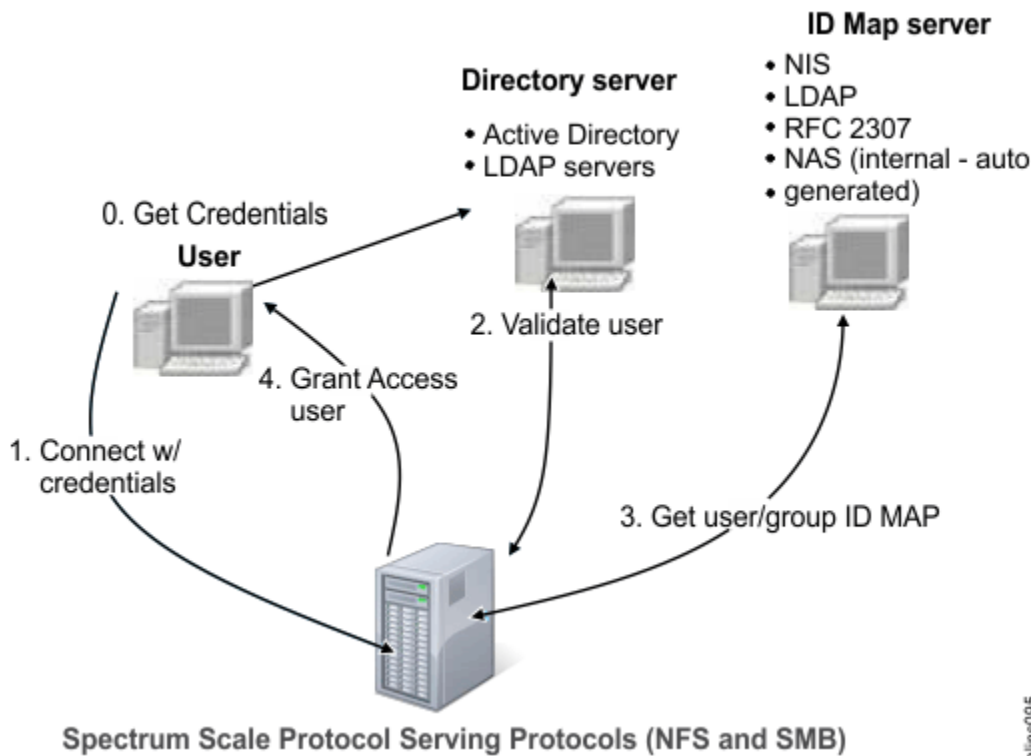


Figure 36. High-level flow of authentication for File protocols

ID mapping

The authentication of the user or groups of users is also associated with the identification of their unique identifiers. To support data access to Microsoft Windows clients (SMB protocol) and to allow interoperability, that is, to share data among UNIX and Windows clients (SMB and NFS protocols), the IBM Spectrum Scale system must map Windows SID to UNIX UID/GID. This process is referred to as ID mapping and the map is referred to as ID map. The ID mapping can be done either internally in the IBM Spectrum Scale system or in an external authentication server.

ID mapping is part of the user identification process in user authentication. The purpose of identification is to identify users and infrastructure components. Identification methods include unique user identifiers (IDs), keys, or fingerprints such as a public Secure Shell (SSH) key, and digital certificates such as a certificate of a web server.

UNIX based systems such as the IBM Spectrum Scale system use user names and user identifiers (UIDs) to represent users of the system. The user name is typically a human-readable sequence of alphanumeric characters and the UID is a positive integer value. When a user logs on to a UNIX system, the operating system looks up the UID and then uses this UID for further representation of the user. User names, UIDs, and the mapping of user names to UIDs are stored locally in the `/etc/passwd` file or on an external directory service such as Active Directory (AD), Lightweight Directory Access Protocol (LDAP), Keystone, or Network Information Service (NIS).

UNIX systems implement groups to maintain sets of users that have the same group permissions to access certain system resources. Similar to user names and UIDs, a UNIX system also maintains group names and group identifiers (GID). A UNIX user can be a member of one or more groups, where one group is the primary or default group. Group names, GIDs, the mapping of group names to GIDs, and the memberships of users to groups are stored locally in the `/etc/group` file or on an external directory service such as Active Directory, LDAP, Keystone, or NIS. The primary group of a user is stored in `/etc/passwd` or in an external directory service.

Windows systems reference all operating system entities as resources. For example, users, groups, computers, and so on are Windows resources. Each resource is represented by a security identifier (SID). Resource names and SIDs are stored locally in the Windows registry or in an external directory service.

such as Active Directory or LDAP. The following methods are used to map Windows SID to UNIX UID and GID:

- External ID mapping methods
 - RFC2307 when AD-based authentication is used
 - LDAP when LDAP-based authentication is used
- Internal ID mapping method
 - Automatic ID mapping when AD-based authentication is used

External ID mapping

A UID or GID of a user or group is created and stored in an external server such as Microsoft Active Directory, NIS server, or LDAP server.

External ID mapping is useful when user UID or group GID is preexisting in the environment. For example, if NFS client with UID and GID as 1000 exists in the environment, and you want a certain share to be accessed by both SMB and NFS client, then you can use an external ID mapping server, assign UID/GID 1000 to the SMB user, and thus, allow both SMB and NFS client to access same data.

Note: The external server administrator is responsible for creating or populating the UID/GID for the user/group in their respective servers.

The IBM Spectrum Scale system supports the following servers for external ID mapping:

- LDAP server, where the UID or GID is stored in a dedicated field in the user or group object on the LDAP server.
- AD server with RFC2307 schema extension defined. The UID or GID of a user or group that is defined in AD server is stored in a dedicated field of the user or group object.

The UID/GID defined in external server can be used by the IBM Spectrum Scale system.

LDAP ID mapping is supported only when the IBM Spectrum Scale is configured with LDAP authentication.

With the external ID mapping methods, a user is always mapped to a *uid* and a group is always mapped to a *gid*. This matches the user and group management in POSIX systems. It carries the limitation that a file or a directory can never be owned by a group. In Microsoft Windows and SMB semantics, it is possible for objects to be owned by a group. However, in IBM Spectrum Scale and an external ID mapping method, that cannot be supported.

Internal ID mapping

The UID or GID of a user or group is created automatically by the IBM Spectrum Scale system and stored in the internal repositories.

When an external ID mapping server is not present in the environment or cannot be used, the IBM Spectrum Scale system uses its internal ID mapping method to create the UID/GID.

IBM Spectrum Scale supports an Automatic ID mapping method if AD-based authentication is used. The Automatic ID mapping method uses a reserved ID range to allocate an ID based on the following logic: A user or group in AD is identified by SID, which includes a component that is called RID. Whenever a user or group from an AD domain accesses IBM Spectrum Scale, a range is allocated per AD domain. The UID or GID is then allocated depending upon this range and the RID of the user/group.

Internal ID mapping cannot be used when the user UID or group GID is preexisting in the environment. However, while using internal ID mapping, if an NFS client wants to access data, then the NFS client must have a UID or GID that is identical to the one created by the IBM Spectrum Scale system.

With the internal ID mapping method, both a *uid* and a *gid* are assigned to the same object at the same time. Querying both even reports user and group information for the same object. That allows to follow SMB semantics more closely, as users and groups are treated very similar in many aspects. This also enables to have an object owned by a group.

Other supported authentication elements for file access

The system supports the following authentication elements as well:

- **Netgroups:** Groups of hosts are used to restrict access for mounting NFS exports on a set of hosts, and deny mounting on the remainder of the hosts. The IBM Spectrum Scale system supports only the netgroups that are stored in NIS and in Lightweight Directory Access Protocol (LDAP).
- **Kerberos:** Kerberos is a network authentication protocol that provides secured communication by ensuring passwords are not sent over the network to the system. The system supports Kerberos with both AD and LDAP-based authentication. When you configure AD-based authentication for any ID mapping method (automatic, RFC2307, LDAP), the Kerberos is enabled for the SMB access by default. However, Kerberos for NFS access is supported only for RFC2307 ID mapping method in AD-based authentication, and to enable NFS Kerberos access for AD-based authentication with RFC2307 ID mapping; you need to use the `--enable-nfs-kerberos` and `--unixmap-domains` options in the **mmuserauth** command.

Kerberos is optional for both SMB and NFS access in LDAP. To enable Kerberos with LDAP, you need to integrate the system with MIT KDC.

- **Transport Level Security (TLS):** The TLS protocol is primarily used to increase the security and integrity of data that is sent over the network. These protocols are based on public key cryptography and use digital certificates based on X.509 for identification.

Caching user and user group information

Every UID and GID, whether generated automatically or through RFC2307 or NIS, is cached during a login attempt. For each successful user login attempt, the UID and GID are stored for seven days. For each failed user login attempt, the UID and GID are cached for two minutes.

When using an external ID mapping server, it is recommended that you not to change any previously created UID or GID. If you must change the UID or GID, plan this activity carefully, preferably before any data associated with the UID or GID exists on the IBM Spectrum Scale cluster. After the change, ensure that the cached entries are flushed out.

Caching user and user group information while using LDAP-based authentication

In LDAP-based user authentication, the **mmuserauth service create** command populates the LDAP bind user and bind password in its configuration file that is located at: `/etc/pam_ldap.conf`. Using this bind user and password, the *username* is looked up in the LDAP server with the configured *login_attribute*. This is the *uid* by default. As a result, *uid=<username>* is looked up in LDAP to fetch the DN for this user. If found, the DN and the *password* are used to perform an LDAP bind operation. The authentication fails if either the *username* is not found or if the bind operation fails.

Note: If LDAP is used for UNIX style login attempts, the *username* is also compared with what is returned from the LDAP server.

For SMB authentication, the ability to perform the LDAP bind with 'username' and 'password' are not available, since the password is in clear text. Therefore, the SMB client passes an NT hash for the password. The Samba server compares the NT hash from the client to the associated NT hash fetched from LDAP server. For this, the **mmuserauth service create** command configures the Samba registry with the LDAP bind user and password. The LDAP bind user and password are used to look up the Samba-related information from the LDAP server, such as the SID and the NT hash. If the NT hashes match for the bind user, then system access is granted.

User credentials, group ID mapping, and group membership caching: User credentials are not cached in the IBM Spectrum Scale system. The user and group ID mapping, and the group membership cache information, are stored in the SSSD component. The cache, local to each node, is for 90 minutes. There is no IBM Spectrum Scale native command to purge this cache. However, the `sss_cache` command (from the `sss-common` package) can be used to refresh the cache.

SMB data caching: There is a 7 day cache period for the user/group SID within the Samba component of the IBM Spectrum Scale system. This cache is local to each node. There is no IBM Spectrum Scale native

command to manage this cache. However, the 'net cache flush' command may be used, on a per node basis, to purge the cache. A negative cache entry persists for 2 minutes.

Netgroup caching: The netgroup information from LDAP is also cached for 90 minutes with the SSSD component. The `sss_cache` command can be used to refresh the cache.

Caching user credentials while using NIS-based authentication for file access

In IBM Spectrum Scale, an NIS server may be used only for users and groups, UID/GID, and group membership lookups for NFS access. In addition, the netgroups defined in NIS are honored as NFS client attributes within the NFS export configuration. Users and groups, UID and GID information, names, and group membership are cached within the SSSD component. This cache, local to each node, is for 90 minutes. There is no IBM Spectrum Scale native command to purge this cache.

The `sss_cache` command from the `sssd-common` package may be used to refresh the cache. The `sss_cache` utility only marks the entries in the respective database as expired. If the NIS server is online and available, the expired entries are refreshed immediately. If the NIS server is not available, the old cached entries that are still marked expired become valid. These values are refreshed once the server is available.

The netgroup information from the NIS is also cached for 90 minutes within the SSSD component. The `sss_cache` command can be used to refresh the cache.

Caching user and group ID mapping and group membership details while using AD-based authentication

The Samba component in the IBM Spectrum Scale system uses the SMB protocol NTLM and Kerberos authentication for user authentication. User and group SID to UID and GID mapping information is cached within the Samba component. This cache, local to each node, is maintained for seven days. There is no IBM Spectrum Scale native command to manage this cache. However, the `net cache flush` command may be used to refresh the cache. The negative cache entry persists for 2 minutes.

The group membership cache, local to each IBM Spectrum Scale protocol node, lies within the Samba component. For an authenticated user, its group membership cache is valid for the lifetime of that session. The group membership is only refreshed on a new authentication request. Therefore, if an SMB tree connect is requested on an existing session, the group cache is not refreshed. So, if there are group membership changes made on the AD server, all of the existing sessions for that user must be disconnected in order to obtain a new authentication request and subsequent cache refresh for the user. If there is no new authentication request made on behalf of the user, a simple user and group information look-up is requested. For example, if you issue the `id` command on the protocol node, the winbind component in IBM Spectrum Scale searches the database to check whether that user's cached information exists from a previous authentication request. If an appropriate entry is found, it is returned. Otherwise, the winbind fetches the mapping information from the AD server and caches it for five minutes.

Note: If a user was authenticated in the past, its group membership cache information is within the database and valid for the session lifetime. Winbind keeps referring to this information and will never try to fetch new information from the AD server. To force winbind to fetch new information, you need to make an authentication request, on behalf of the user, to the node in the same way as you would connect from a CIFS client.

Related concepts

Impacts of authentication on enabling and disabling protocols

The following are the recommendations for enabling and disabling protocols.

Authentication for object access

The OpenStack identity service that is enabled in the system confirms an incoming request by validating a set of credentials that are supplied by a user. The identity management consists of both authentication and authorization processes.

Deleting authentication and ID mapping

You can choose to delete the ID mappings that are generated and stored on IBM Spectrum Scale. The authentication method that is configured in the system can be deleted too. This deletion might result in a

complete loss of data access. Before you delete an ID mapping, determine how to maintain data access if needed.

Authentication for object access

The OpenStack identity service that is enabled in the system confirms an incoming request by validating a set of credentials that are supplied by a user. The identity management consists of both authentication and authorization processes.

In the authentication and authorization process, an object user is identified in the IBM Spectrum Scale system by attributes such as user ID, password, project ID, role, and domain ID with Keystone API V3. The keystone server that is used for the OpenStack identity service that is configured on the IBM Spectrum Scale system manages the user authentication requests for object access with the help of either an external or internal authentication server for the user management. You can configure an internal or an external keystone server to manage the identity service. You can use the following authentication methods for object access either by using an internal keystone server that ships IBM Spectrum Scale or by using an external keystone server:

External authentication

The keystone server interacts with the following external servers for the user management:

- AD
- LDAP

Internal authentication

The keystone server interacts with an internal database to manage users and authentication requests.

The user can select the authentication method based on their requirement. For example, if the enterprise deployment already consists of AD and the users in the AD need to access object data, the customer can use AD as the backend authentication server.

When the authentication method selected is either AD or LDAP, the user management operations such as creating a user and deleting a user are the responsibility of the AD and LDAP administrator. If local authentication is selected for object access, the user management operations must be managed by the keystone server administrator.

The authorization tasks such as defining user roles, creating projects, and associating a user with a project are managed by the keystone administrator. The keystone server administration can be done either by using Keystone V3 REST API or by using OpenStack python-based client that is called `openstackclient`.

The following diagram shows how the authentication requests are handled when IBM Spectrum Scale is with an internal keystone server and external AD or LDAP authentication server.

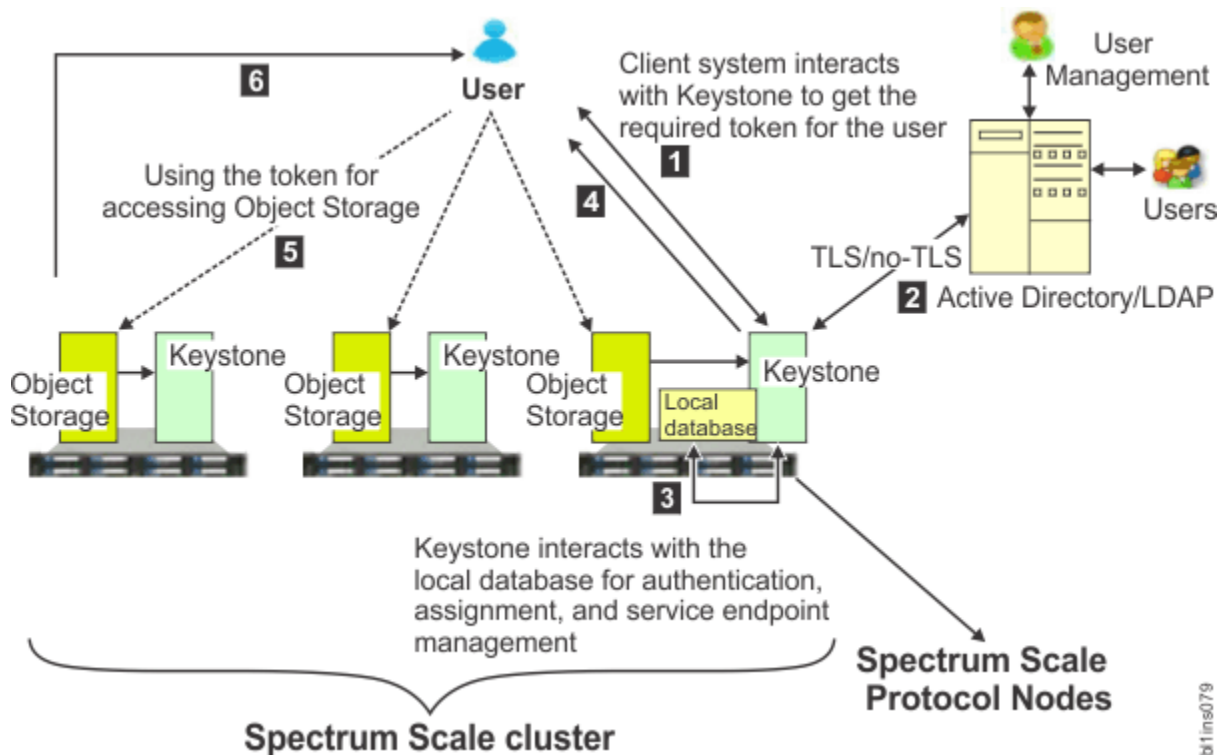


Figure 37. IBM Spectrum Scale integration with internal Keystone server and external AD or LDAP authentication server

The following list provides the authentication process for object access:

1. The user raises the access request to get access to the object data.
2. The keystone server communicates with the authentication server (such as AD, LDAP, or a local database). The keystone server interacts with the authentication server for authentication, authorization, and service end-point management.
3. If the user details are valid, the keystone server interacts with the local database to determine the user roles and issues a token to grant access to the user.
4. The OpenStack identity service offers token-based authentication for object access. When user credentials are validated, the identity service issues an authentication token, which the user provides in subsequent requests. That is, the access request also includes the token that is granted in step 3. The token is an alphanumeric string of text that is used to access OpenStack APIs and resources.
5. The authenticated user contacts the object storage to access the data that is stored in it.
6. The object storage grants permission to the user to work on the data based on the associated project ID and user role.

Each object user is part of a project. A project is used to group or isolate resources. Each user needs to be defined with the set of user rights and privileges to perform a specific set of operations on the resources of the project to which it belongs to.

The Identity service also tracks and manages the access to the OpenStack services that are installed on the system. It provides one or more endpoints that can be used to access resources and perform authorized operations. Endpoints are network-accessible addresses (URLs) that can be used to access the service.

When the user is authenticated, the keystone server provides a list of services and a token to the user to access the services. For example, if the user is authenticated to access the Object Storage service, the keystone server provides the token to access the service. The Object Storage service then verifies the token and fulfills the request.

To achieve high availability, Object Storage and Keystone are activated on all protocol nodes. The internal database that is required to validate the user is installed on the shared root file system.

Depending upon the configuration, the keystone server needs to interact with AD, LDAP, or the internal database for user authentication. The Keystone internal database is also used for assigning users to projects with a specific role for controlling access to projects. It also holds the definition of OpenStack services and the endpoints for those services.

Each node that is configured with object storage is configured to interact with the Keystone server.

Related concepts

[Impacts of authentication on enabling and disabling protocols](#)

The following are the recommendations for enabling and disabling protocols.

[Authentication and ID mapping for file access](#)

The system supports an external authentication service to authenticate users on the system. Before you configure an authentication method, ensure that the external authentication service is set up correctly.

[Deleting authentication and ID mapping](#)

You can choose to delete the ID mappings that are generated and stored on IBM Spectrum Scale. The authentication method that is configured in the system can be deleted too. This deletion might result in a complete loss of data access. Before you delete an ID mapping, determine how to maintain data access if needed.

Deleting authentication and ID mapping

You can choose to delete the ID mappings that are generated and stored on IBM Spectrum Scale. The authentication method that is configured in the system can be deleted too. This deletion might result in a complete loss of data access. Before you delete an ID mapping, determine how to maintain data access if needed.

It is not possible to delete both an authentication method and the current ID mappings at the same time. If you need to delete the current ID mappings, you need to first delete the authentication method that is configured in the system. If only the authentication settings are deleted, you can return to the existing setting by reconfiguring authentication. You cannot recover a deleted ID mapping. Typically, IDs are deleted in test systems or as part of a service operation.

For more information about how to delete authentication and ID mappings, see *Deleting authentication and ID mapping configuration* in *IBM Spectrum Scale: Administration Guide*.

Related concepts

[Impacts of authentication on enabling and disabling protocols](#)

The following are the recommendations for enabling and disabling protocols.

[Authentication and ID mapping for file access](#)

The system supports an external authentication service to authenticate users on the system. Before you configure an authentication method, ensure that the external authentication service is set up correctly.

[Authentication for object access](#)

The OpenStack identity service that is enabled in the system confirms an incoming request by validating a set of credentials that are supplied by a user. The identity management consists of both authentication and authorization processes.

Planning for NFS

You must make a number of decisions before you deploy NFS in an IBM Spectrum Scale environment. An outline of these decision points is as follows.

The IBM Spectrum Scale for NFS architecture is shown in [Figure 38 on page 302](#).

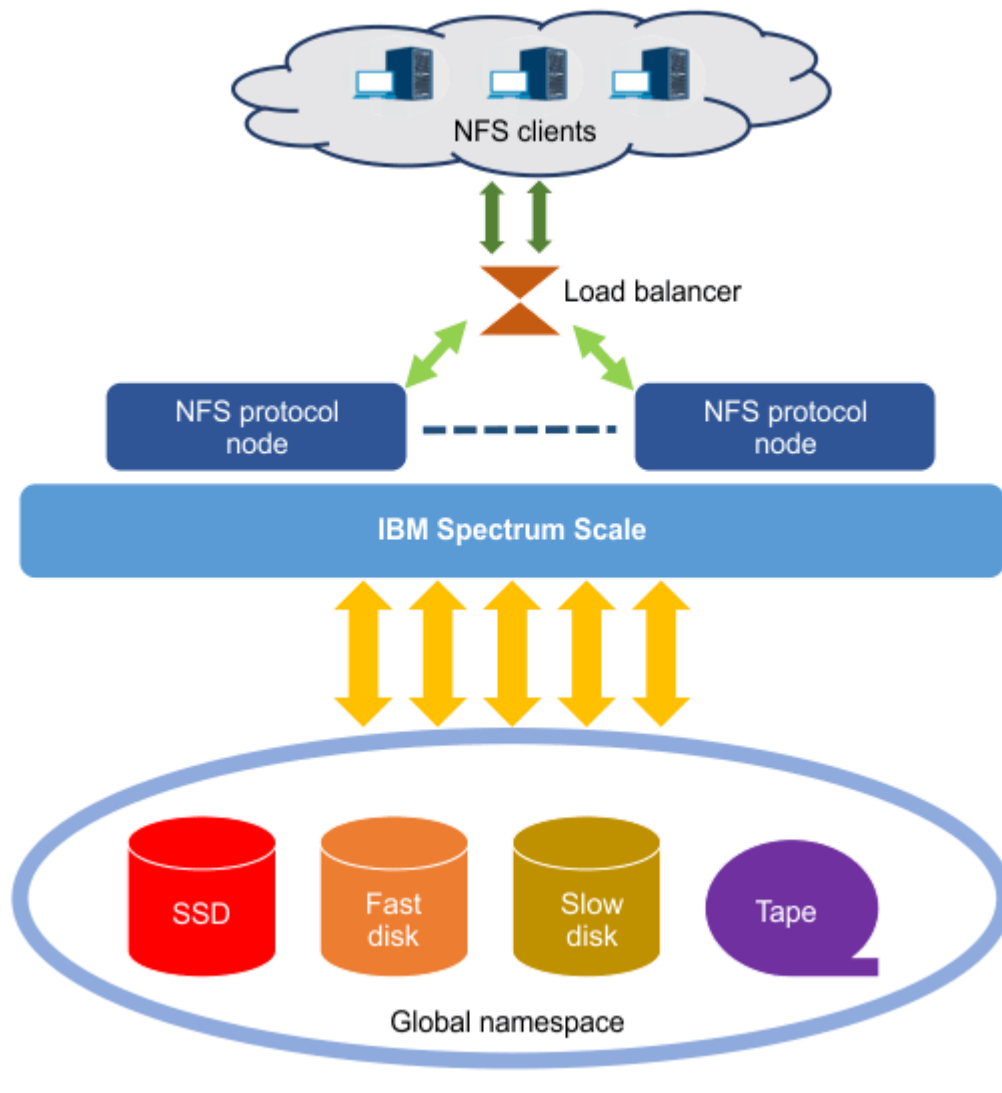


Figure 38. IBM Spectrum Scale for NFS architecture

Each IBM Spectrum Scale protocol node that has an enabled NFS, runs the NFS service, other protocol services (SMB and Object, if enabled), and the IBM Spectrum Scale client. NFS users make requests through an NFS client to perform an NFS operation on a mounted NFS file system, for example, to read or to write a file. The request is routed to an NFS protocol node. The NFS server on the protocol node handles the request against the backend IBM Spectrum Scale storage file system. The request completion status is then returned to the user through the NFS client.

File system considerations for the NFS protocol

Ensure that all IBM Spectrum Scale file systems, which export data via NFS, are mounted with the `syncnfs` option to prevent clients from running into data integrity issues during failover.

Linux operating systems

On Linux operating systems, the IBM Spectrum Scale `syncnfs` setting is the default mount option.

AIX operating system

On AIX, the IBM Spectrum Scale `nosyncnfs` setting is the default mount option.

Enter the following command to change the setting to `syncnfs`:

```
# mmchfs fileSystem -o syncnfs
```

Enter the following command to verify:

```
# mmlsfs fileSystem -o
```

For more information about mount options, see *Mount options specific to IBM Spectrum Scale* in *IBM Spectrum Scale: Command and Programming Reference*.

For fileset considerations for the NFS protocol, see [“Fileset considerations for creating protocol data exports”](#) on page 315.

NFS tested clients

IBM Spectrum Scale tested some clients that can be used for accessing IBM Spectrum Scale data by using NFS protocols.

IBM support can help resolve NFS client/server issues when these issues can be re-created on one of the NFS clients that are listed here as tested.

Table 30. Tested NFS clients					
Operating System	PPC LE	PPC BE	X86_64	X86_32	s390x
RHEL6	✓	X	✓	X	
RHEL7	✓	X	✓	X	✓
RHEL8	✓		✓		✓
SLES12		X	✓		✓
SLES15			✓		✓
AIX6		X			
AIX7		✓			
z/OS® V2					X
Ubuntu 20.04	✓		✓		

You must always:

- Install the latest fix level for the operating system.
- Run an operating system still supported by the vendor.
- Keep IBM Spectrum Scale on the latest fix level.

The following clients are not explicitly supported:

- The blank cells in [Table 30 on page 303](#) denote an unsupported environment.
- All versions of Windows NFS client.

Considerations for NFS clients

When you use NFS clients in an IBM Spectrum Scale environment, the following considerations apply.

- If you mount the same NFS export on one client from two different IBM Spectrum Scale NFS protocol nodes, data corruption might occur.
- IBM Spectrum Scale allows concurrent access to the same file data by using SMB, NFS, and native POSIX access. For concurrent access to the same data, some limitations apply. For more information, see *Multiprotocol export considerations* in *IBM Spectrum Scale: Administration Guide*.
- 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 version in the **mount** command. For

more information, see the **mount** command for your client operating system. If you are planning to use NFSv4.1 version, ensure that the client supports the version.

- It is recommended to use the CES IP address of the IBM Spectrum Scale system to mount the NFS export on an NFS client.
 - You can use the **mmces address list** command to view the CES IPs.
 - You can use the **mmclscluster --ces** command to determine which nodes host which CES IPs.
- For items that might affect NFS access, see the *Authentication limitation* topic in the *IBM Spectrum Scale: Administration Guide*.

CES NFS limitations

This topic describes the limitations of IBM Spectrum Scale CES NFS protocol.

IBM CES NFS stack limitations

CES NFS limitations are described here:

- Changes to the IBM CES NFS global configuration are not dynamic. NFS services automatically restart during the execution of the **mmnfs export load** and **mmnfs config change** commands. During this time, an NFS client with a soft mount might lose connectivity. This might result in an application failure on the client node. An NFS client with a hard mount might "stall" during the NFS restart.
- Whenever NFS is restarted, a grace period will ensue. The NFS grace period is user configurable, and the default NFS grace period is 90 seconds. If NFS global configuration changes are performed sequentially, then NFS services will be restarted multiple times, leading to a cumulative extended grace period. This might prevent NFS clients from reclaiming their locks, possibly leading to an application failure on the client node.
- NFS connections are limited to a maximum of 2250 for a large number of NFS exports.
- The maximum number of NFS exports supported per protocol cluster is 1000.
- Exporting symbolic links is not supported in CES NFS.

NFS protocol node limitations

When mounting an NFSv3 file system on a protocol node, the Linux kernel **lockd** daemon registers with the rpcbind, preventing the CES NFS lock service from taking effect. If you need to mount an NFSv3 file system on a CES NFS protocol node, use the **-o nolock** mount option to prevent invoking the Linux kernel **lockd** daemon.

NFS Ganesha protocol server's client limitation

Microsoft Windows are not supported and tested client for CES NFS Ganesha protocol server. If you want to use Windows as client, CES SMB is the recommended protocol.

Limitations while using nested exports with NFS

Creating nested exports (such as `/path/to/folder` and `/path/to/folder/subfolder`) is not recommended as this might lead to serious issues in data consistency. Remove the higher-level export that prevents the NFSv4 client from descending through the NFSv4 virtual filesystem path. In case nested exports cannot be avoided, ensure that the export with the common path, called as the top-level export, has all the permissions for this NFSv4 client. Also, NFSv4 client that mounts the parent (`/path/to/folder`) export does not see the child export subtree (`/path/to/folder/inside/subfolder`) unless the same client is explicitly allowed to access the child export as well.

NFS export considerations for versions prior to NFS V4

For NFS exported file systems, the version of NFS you are running with may have an impact on the number of inodes you need to cache, as set by both the `maxStatCache` and `maxFilesToCache` parameters

on the `mmchconfig` command. The performance of the `ls` command in NFS V3 in part depends on the caching ability of the underlying file system. Setting the cache large enough will prevent rereading inodes to complete an `ls` command, but will put more of a CPU load on the token manager.

Also, the clocks of all nodes in your GPFS cluster must be synchronized. If this is not done, NFS access to the data, as well as other GPFS file system operations, may be disrupted.

NFS V4 export considerations

For information on NFS V4, refer to *NFS Version 4 Protocol* and other information found in the [Network File System Version 4 \(nfsv4\) section of the IETF Datatracker website \(datatracker.ietf.org/wg/nfsv4/documents\)](http://datatracker.ietf.org/wg/nfsv4/documents).

To export a GPFS file system using NFS V4, there are two file system settings that must be in effect. These attributes can be queried using the `mm lsfs` command, and set using the `mmcrfs` and `mmchfs` commands.

1. The `-D nfs4` flag is required. Conventional NFS access would not be blocked by concurrent file system reads or writes (this is the POSIX semantic). NFS V4 however, not only allows for its requests to block if conflicting activity is happening, it insists on it. Since this is an NFS V4 specific requirement, it must be set before exporting a file system.

flag	value	description
-D	nfs4	File locking semantics in effect

2. The `-k nfs4` flag is required. To export a file system by using NFS V4, NFS V4 ACLs must be enabled. Since NFS V4 ACLs are vastly different and affect several characteristics of the file system objects (directories and individual files), they must be explicitly enabled. This is done by specifying `-k nfs4`.

flag	value	description
-k	nfs4	ACL semantics in effect

3. For NFS users with more than 16 groups, set `MANAGE_GIDS=TRUE`, else user can not get access to NFS exports.

Planning for SMB

Several steps need to be taken before the SMB service is implemented in IBM Spectrum Scale.

The SMB support for IBM Spectrum Scale allows clients to access the GPFS file system by using SMB clients. The SMB service on the protocol node provides file serving to SMB clients. Requests are routed to an SMB protocol node, typically by using DNS round robin. The SMB server on the protocol node handles these requests by issuing calls to the IBM Spectrum Scale file system.

Note: The SMB service is based on a clustering component CTDB. You must have the CTDB network traffic on some dedicated private network. The CTDB traffic is using the same network interface as the GPFS daemon. It can be configured or changed through the `--daemon-interface` option in the `mmaddnode` and `mmchnode` commands. The recommendation is to ensure that this is on a private network that is not visible outside the cluster.

SMB connections

Each IBM Spectrum Scale protocol node is capable of handling a large number of SMB connections.

The number of concurrent SMB connections that a protocol node can handle is dependent on the following factors:

- Number of processors
- Amount of memory installed in the protocol node
- The IO workload. This depends on the following factors:
 - Number of concurrent SMB connections

- Lifetime and frequency of SMB connections
- Overhead due to metadata operations
- Frequency of operations on each SMB connection
- Concurrent access to the same files
- The storage configuration and advanced functions that are configured to run while serving high number of SMB connections

SMB fail-over scenarios and upgrade

When you are planning an IBM Spectrum Scale system configuration that has a high number of SMB connections, you must consider the impact that a fail-over can have on the performance of the system.

When an IBM Spectrum Scale protocol node fails, the IP addresses that are hosted by that protocol node are moved to another IBM Spectrum Scale protocol node. SMB clients must reconnect to one of the remaining CES nodes. After the fail-over is processed, the same IP address can be used. The remaining protocol nodes handle all the SMB connections.

Therefore, when you plan an IBM Spectrum Scale system that has a high number of SMB connections, some buffer, in terms of number of SMB connections, must be factored into the overall system configuration. This contingency buffer prevents a system overload during a fail-over scenario, thus reducing any adverse effects on the system performance.

A similar consideration applies to SMB upgrades. The SMB upgrade happens in two phases. During the first phase of the upgrade process, the first half of the nodes are updated. The remaining nodes handle the SMB connections of the updated node. When the first half of the nodes are updated, the upgrade moves to the second phase. In the second phase, SMB is shut down completely. This practice is followed to update the SMB code on all of the remaining protocol nodes concurrently. However, it leads to a brief outage of the SMB service. For more information, see [“Upgrading SMB packages” on page 521](#).

SMB limitations

This topic describes the limitations of IBM Spectrum Scale SMB protocol.

IBM Spectrum Scale allows concurrent access to the same data file through SMB and NFS, and through native POSIX access.

For more details on these options, see the *mmsmb* command in the *IBM Spectrum Scale: Command and Programming Reference*.

For more information on SMB client access and limitations, see *Multiprotocol export considerations* in the *IBM Spectrum Scale: Administration Guide*.

SMB server limitations

The SMB server does not indicate mount points to the SMB client. This can have the effect that SMB clients query free space in the wrong directory, if different filesets are linked to subfolders in an SMB share. When an SMB CREATE request for creating or replacing a file fails, an empty file can be left behind.

SMB share limitations

Consider all the limitations and support restrictions before you create an SMB share.

SMB share limitations include the following:

- NTFS alternate data streams are not supported. For example, named streams generated by a Mac OS X operating system cannot be stored directly.

Enabling *vfs_fruit* will partially lift this limitation. For more information, see the *Support of vfs_fruit for the SMB protocol* topic in the *IBM Spectrum Scale: Administration Guide*.

- The encryption status of files cannot be queried or changed from SMB clients. Use the **CLI** commands instead.

- When propagation of opportunistic locks across protocols is enabled (SMB option gpfs:leases), then Level 2 oplocks are not granted and Exclusive or batch oplocks are not broken down to Level 2 oplocks and are revoked from the system.
- Concurrent access to the same files or directories from many SMB connections on multiple nodes can result in scalability problems.
- Symbolic links cannot be created or changed from SMB clients and are not reported as symbolic links.
- Symbolic links created via NFS or directly in the file system will be respected as long as they point to a target under the same shared directory.
- Windows Internet Name Service (WINS) is not supported.
- Retrieving Quota information using NT_TRANSACT_QUERY_QUOTA is not supported.
- Setting Quota information using NT_TRANSACT_SET_QUOTA is not supported.
- Setting the maximum number of connections to a share is not supported. The MMC GUI allows specifying this parameter, but it cannot be set on an IBM Spectrum Scale cluster.
- UNIX Extensions are not supported.
- You cannot create a shadow copy of a shared folder using a remote procedure call from a shadow copy client. Backup utilities, such as Microsoft Volume Shadow Copy Service, cannot create a shadow copy of a shared folder using a procedure call.
- The Branch Cache hash operations using SRV_READ_HASH IOCTL are not supported.
- Leases are not supported.
- Only the SMB2 and SMB3 protocol versions are supported.
- No support of dynamic ACLs and SACLs.
- SMB Direct (SMB over RDMA) is not supported.
- Multi-channel for SMB is not supported.
- Durable handles are not supported.
- Resilient handles are not supported.
- Persistent handles are not supported.
- Storing arbitrary data in Security Descriptors is not supported.
- Pre-allocating space larger than the file size is not supported. A check for the available disk space is used instead when processing an allocation request from an SMB client.
- The Witness notification protocol is not supported.
- Continuous Availability (CA) SMB exports are not supported.
- Scaleout SMB exports are not supported.
- Creating snapshots from an SMB client is not supported.
- Application instance ID is not supported.
- FSCTL_SET_ZERO_DATA is only supported on sparse files.
- Setting compression from an SMB client or querying the compression state from an SMB client is not supported.
- All CES nodes need to be the same hardware architecture (x86 versus Power) and the same endianness (little endian versus big endian).
- Rolling code upgrade is not supported for SMB. The SMB service needs to be stopped on all CES nodes for the upgrade to ensure that the same SMB software level is used on all CES nodes.
- When upgrading IBM Spectrum Scale from version 4.2.3.19 to version 5.0.5, the max protocol level for SMB support does not automatically change to the latest. This has to be done manually.
- The AAPL create context used by Mac OS clients is not supported. When enabling VFS fruit AAPL create contexts are supported.
- The number of open files for each SMB connection is limited to 16384.

- IBM Spectrum Scale can host a maximum of 1,000 SMB shares. There must be less than 3,000 SMB connections per protocol node and less than 20,000 SMB connections across all protocol nodes.
- No support of SID history.
- The SMB service requires file names and directory names to use the UTF-8 encoding. Files and directories with different encoding cannot be accessed through the SMB protocol.
- The NFSv4 SYNCHRONIZE bit is no longer automatically set on file ACLs. Windows clients expect SYNCHRONIZE on ACLs to correctly allow rename.
- 8.3 short name creation is not supported
- IBM Spectrum Scale can act as a MS-DFS target or can redirect an entire share. But it does not allow any redirects from within an exported tree.

SMB node limitations

IBM Spectrum Scale allows a maximum of 16 SMB server nodes in a CES cluster.

SMB user limitations

When a new security feature is added for CVE-2020-25717, Samba by default blocks access for users having UNIX uid below 1000. This was done to enhance the samba security. Additionally, a new **min domain uid** parameter (default 1000) has been added and no matter how we obtain the UNIX uid to use in the process token (we may eventually read `/etc/passwd` or similar), by default no UNIX uid below this value will be accepted.

If user wants the UIDs below 1000 to work, they must set **min domain uid** to the lowest UNIX user uid that they contain.

Important: At present, this **min domain uid** parameter is not supported through **mmsmb** command. Therefore, the lowest UNIX user uid can be set by using **samba net** command.

The following sample command shows usage of **samba net** command to set the **min domain uid** parameter value:

```
/usr/lpp/mmfs/bin/net conf setparm global 'min domain uid' 500
```

Here, 500 is the minimum Unix uid of the user that need SMB access.

For information on CVE - 2020 - 25717 security feature, see [Samba security CVE-2020-25717](#).

SMB encryption on Power systems

SMB encryption on Power systems requires Openssl 1.0.2 or newer for crypto-hardware acceleration, which is provided by RHEL 7.4 or newer. Also, the hardware crypto-acceleration is only available on POWER8® or newer systems.

SMB data migration to IBM Spectrum Scale

This topic documents a procedure for migrating data from an SMB system to IBM Spectrum Scale while preserving SMB metadata like ACLs and SMB file attributes.

Prerequisites for SMB data migration

This topic describes the prerequisites that must be met before you go ahead with SMB data migration

The recommended tool for migration is Robocopy, which should be running from one or more Windows client machines. The data is read through the SMB protocol from the source machine and written through the SMB protocol to IBM Spectrum Scale file system.

The prerequisites are the following:

- Authentication method has to be Active Directory with the source SMB server and target IBM Spectrum Scale cluster using the same Active Directory environment.

- Windows client machines for the data migration can access the source SMB server and the IBM Spectrum Scale cluster
- A user ID used for the migration needs to be defined and that user must have permissions to read all data from the source SMB server.
- SID history needs to be removed or converted on the source data.

Setting up the IBM Spectrum Scale environment for capacity and performance

The first step for data migration is to set up the IBM Spectrum Scale nodes if you have not already done so. For more information, see [Chapter 3, “Steps for establishing and starting your IBM Spectrum Scale cluster,”](#) on page 349.

To achieve optimum performance while migrating small files, ensure the following:

- Use many Windows clients to scale out the number of SMB sessions
- Use Robocopy with /MT option
- Disable the syncops: onclose option while creating the SMB exports, if applicable

Configuring IBM Spectrum Scale authentication

The authentication method to be used is Active Directory (AD) authentication. For more information about setting up file authentication, see the **mmuserauth service create** section of the **mmuserauth** command in the *IBM Spectrum Scale: Command and Programming Reference*.

Note: Robocopy reads SIDs from the source SMB server, sends them to IBM Spectrum Scale. IBM Spectrum Scale creates new UNIX UIDs and GIDs based on the ID mapping that is configured on the IBM Spectrum Scale CES cluster.

Creating SMB exports

Create SMB exports for the target of the data migration. For more information on creating SMB exports, see the **mmsmb** command in the *IBM Spectrum Scale: Command and Programming Reference*. Ensure to configure the migration user ID as "admin user" for all SMB exports. See the **mmsmb export change SMBexport --option 'admin users=domain\user' options**.

Note: The user identifier used for the migration should have rights to set file ownership and ACLs.

Accessing the source SMB server and the IBM Spectrum Scale file system

Do the following:

1. From the Windows clients, access the source SMB server and the IBM Spectrum Scale cluster.
2. Map network drives for the source SMB server and the IBM Spectrum Scale cluster.

If data for multiple SMB exports are transferred, multiple Windows clients can be used, where each one accesses one pair of source and target SMB exports. This allows to speed up the transfer.

Running the Robocopy commands

For each SMB export that you want to migrate, run the **Robocopy** command as follows:

```
Robocopy \\<source-smb-server>\<smbexport> \\<target-scale-cluster>\<smbexport> /copy:ATSO /secfix /Z /E /MT:32 /R:5 /W:3 /s1 /log:logfile .
```

The Robocopy options are as follows:

- /COPY:ATSO: Copy file information.
 - A: Attributes
 - T: Timestamps
 - S: Security - NTFS ACLs

- O: Owner information
- /secfix: Fixes the security on all files
- /Z: Ensures Robocopy can resume the transfer of a large file in mid-file instead of restarting.
- /E : Copy all subdirectories including the empty ones.
- /MT : Create multi-threaded copies with N threads. N must be an integer between 1 and 128. The default value for N is 8.
- /R: Specifies the number of retries on failed copies. The default value is 1,000,000 (one million retries).
- /W: Specifies the wait time between retries, in seconds. The default value is 30 (wait time 30 seconds).
- /sl : Copies the symbolic link instead of the target.
- /log: Writes the status output to the log file (overwrites the existing log file)

Cleanup after migration

Do the following steps after migration:

1. Unmount all SMB exports from the Windows clients.
2. Remove the "admin users" entry as it is no longer required and can be harmful if used by accident.

SMB best practices

Some SMB best practices are highlighted in the following sections.

IBM Spectrum Scale configuration

Learn about the factors that determine the maximum number of active SMB connections.

The two important factors that determine the maximum number of active SMB connections per protocol node are:

- The protocol node hardware
- The underlying storage systems.

Storage planning and configuration

The storage configuration must be carefully planned based on the overall workload characteristics that are intended for IBM Spectrum Scale.

The factors to be considered in planning the storage configuration include:

- Expected response time, during peak and off peak hours.
- Workload profiling to understand the data access requirements, both in terms of throughput (MiB per second) and I/O operations per second (IOPS).
- Plan for adequate storage resources not just for “normal” network file serving but also for advanced functions.

The storage configuration includes the following details:

- Number of storage nodes.
- Storage disk subsystems.
- Number and type of disk drives.
- Total capacity
- Projected IO throughput and IOPS

A file system that resides on fewer disks, or disks that are mapped on to RAID arrays that comprise slower lower speed disk drives, can result in a lower number of active concurrent SMB connections per protocol node.

SMB Options to enable or disable interoperability

Read about options to enable or disable interoperability, that is, the options that control whether file access metadata is forwarded into the file system to make it available to other clients (POSIX, NFS, etc.) as well.

Leases, locking, and sharemodes

If your files are only accessed via SMB clients, the overhead that is incurred while ensuring consistency across multiple NAS protocols can be avoided if you disable the following interoperability options. This is recommended if the data in the file system is only accessed through the SMB protocol and not through other NAS protocols, such as NFS, FTP, HTTPS, or by POSIX clients.

gpfs:leases

Understand the scenarios in which gpfs:leases are enabled or disabled.

gpfs:leases are enabled by default when an SMB export is created. When enabled, it specifies that clients, which access the file over other NAS protocols, can break the opportunistic lock of an SMB client. This feature informs the SMB client when another client is accessing the same file at the same time through a non-SMB protocol.

Disabling this feature provides a slight increase in performance each time that a file is opened. You can only disable this option if the files are exclusively accessed via SMB. You risk data corruption otherwise. If only the SMB layer manages oplocks, the overhead of keeping them in sync with the core file system is saved.

gpfs:leases can be disabled for a particular SMB export by specifying the

```
-option gpfs:leases = no
```

option on the "create" or "change" subcommands in the *mmsmb* command topic in the *IBM Spectrum Scale: Command and Programming Reference*.

posix:locking

Understand the scenarios in which posix:locking is enabled or disabled.

posix:locking is enabled by default when an SMB export is created. When enabled, it determines whether a byte range file control lock is already present on the requested portion of the file when a byte range lock is granted to an SMB client.

Clients accessing the same file that uses another NAS protocol, such as NFS, are able to determine whether an SMB export sets a lock on that file.

If an export is accessed only by an SMB export, disable the inter-protocol level byte-range locking to enhance the serving performance of the SMB export.

Locking is disabled for a particular SMB export at the Inter-protocol level by specifying the

```
-option posix:locking = no
```

option on the "create" or "change" subcommands in the *mmsmb* command topic in the *IBM Spectrum Scale: Command and Programming Reference*.

gpfs:sharemodes

Read about share modes and how they impact SMB export.

The SMB protocol allows an application simultaneous access to a file by defining share modes when it is first opened. The share modes can be in any of the following combinations.

- SHARE_READ
- SHARE_WRITE
- SHARE_DELETE

If no sharemode is specified, all attempts of simultaneous access by another application or client to open a file, in a manner that conflicts with the existing open mode, is denied. Access is denied even if the user has the appropriate permissions that are granted by share and file system access control lists.

The "sharemodes" option is enabled by default when an SMB export is created. When enabled, the share modes that are specified by SMB clients are respected by other NAS protocols. When disabled, it specifies that the share modes are applicable only to access by SMB clients. Clients that use all other NAS protocols are granted or denied access to a file without regard to any share mode defined by an SMB client.

If the export is not being accessed by clients that use other network file protocols (such as NFS), then it is highly recommended that

```
--option gpfs:sharemodes=no
```

option is specified on the **mmsmb export add** or **mmsmb export change** commands.

For more information, see, [Upgrading SMB packages](#).

Note: If your environment requires data sharing over multiple –protocols and these options cannot be disabled, you might not be able to achieve the maximum active SMB connections per node. In that case, consider adding more protocol nodes and increased storage bandwidth.

Creating home directory exports using substitution variables

Guidance to set up SMB exports for efficient home directories

Having many Windows users all concurrently accessing the same SMB export can lead to performance bottlenecks because Windows clients automatically open the root folder of an export when they connect. In a home directory environment, it is recommended that substitution variables be used when creating SMB exports for home directories. For example, home directory exports can be created by using the %U substitution variable that represents the user name on the **mmsmb export add <export> <path with substitution variable> " command (mmsmb export add home /ibm/gpfs0/.../%U)**. For more information, see the *mmsmb* command topic in the *IBM Spectrum Scale: Command and Programming Reference*.

Sharing files and directories among SMB clients - concurrent access to files

Information on performance optimization when performance slows down due to extensive sharing.

If your environment calls for extensive file and directory sharing among many users, such as a large set of department documents, you can experience slowdown in performance. In this type of environment, it is possible to improve the performance based on the user needs. Consider the following options to optimize performance:

- Use the SMB export coherency options that are described in the following section.
- Limit all sharing through a single protocol node, or as few protocol nodes as possible. Restricting SMB connections that export data to a single protocol node helps reduce internal communication among the protocol nodes.
- If possible, distribute workload in sub directories to reduce number of SMB connections that access the same directory at the same time.

SMB export coherency options: fileid:algorithm

IBM Spectrum Scale provides a coherency, that is, fileid:algorithm, option to control data consistency needs for an SMB export. This option applies when an export is being accessed only by SMB clients. When the default value of **fsname** is changed, it helps to improve performance. However, extreme caution must be taken to determine right settings for your data as it impacts data integrity.

The applications must ensure that files or directories are not modified by multiple processes at the same time. For example, reading and writing of the same file does not happen simultaneously by different processes or alternatively, when the application is coordinating all file accesses to avoid conflicts.

The coherency (**fileid:algorithm**) option can be changed for a particular SMB export by specifying the **--option fileid:algorithm = {fsname| hostname | fsname_nodirs | fsname_norootdir}** option on the **mmsmb export add** or **mmsmb export change** commands.

Table 31. Coherency option and description	
Coherency option	Description
fsname:norootdir	Disables synchronization of directory locks for the root directory of the specified export, but keeps lock coherency for all files and directories within and underneath the export root. This option is useful for scenario where large sets of connections are accessing different sub directories with the same export. A typical example is an export that is used for home directories (/ibm/gpfs0/homeroot) which then contains a subdirectory for each user.
fsname:nodirs	Disables synchronization of directory locks across the cluster nodes, but leaves lock coherency that is enabled for files. This option is useful if data sharing is not dependent of the changes to the directory attributes like time stamps, consistent view of the directory contents.
fsname	Enables cross-node lock coherency for both directories and files. This is the default setting.
hostname	Disables cross-node lock coherency for both directories and files. It needs to be used only with applications that guarantee data consistency and all other options to enhance performance are exhausted.

Scheduling advanced functions for data management

Information and tips on data management tasks timings and schedule planning.

In most environments, it is typical to have an off-peak window of time at some point during the day. This time can be used to conduct data management tasks such as nightly backup, snapshots, and asynchronous replication. You need to ensure that you have some time of lower SMB file activity on the server and that this time window is sufficient for the planned advanced functions to complete.

When running advanced functions that require a file system policy scan such as backup, asynchronous replication, policy invocations, Advanced File Management (AFM), or Active Cloud Engine® (ACE) cache pre-population, schedule them sufficiently apart to allow adequate time to complete the policy scan. The planned time gap helps to avoid two overlapping policy scans.

In a typical system, a couple of hours gap between two advanced functions is sufficient. However, you need to review the logs of each function to ensure that the scan completes before the next scheduled advanced function starts. If necessary, make adjustments like increasing the time gap, adding more protocol nodes, more disks for metadata, or adding Solid State Disks (SSDs) for metadata for IBM Spectrum Scale gateway configuration.

As you plan data management tasks for your environment, you need to ensure adequate resources are available to complete all data management tasks. If these tasks do not complete in the expected time window or they impact overall performance of the system during peak hours, consider adding additional resources such as dedicated protocol nodes for backup or extra storage resources to enhance storage response time. This can also eliminate bottlenecks.

Planning for fail-over scenarios and upgrade

Information on performance impact and management during failover scenario of IBM Spectrum Scale protocol node.

During the planning of a system that has a high number of active concurrent SMB connections, the potential performance impact during fail-over scenarios needs to be considered.

If an IBM Spectrum Scale protocol node fails, the IP addresses that are hosted by that protocol node are relocated to other IBM Spectrum Scale protocol nodes, and SMB client reconnections are redistributed among the remaining IBM Spectrum Scale nodes.

When a high number of active concurrent SMB connections are planned, the overall system configuration needs to factor some buffer space in terms of maximum active concurrent SMB connections. This accounts for the potential performance implications during these fail-over scenarios.

During the IBM Spectrum Scale software upgrade process, IP addresses are frequently relocated. Depending on the configuration, multiple protocol nodes are suspended concurrently to minimize the upgrade time. This leaves fewer protocol nodes to serve various protocol clients including SMB. Therefore, the maximum number of active SMB connections can not be sustained during the IBM Spectrum Scale software upgrade process. You need to plan for upgrade during the off-peak hours or schedule a maintenance window to minimize the impact on clients that access IBM Spectrum Scale.

For more information on upgrade planning, see [“SMB fail-over scenarios and upgrade” on page 306](#).

Analysis of performance concerns and fine-tuning

Information on addressing performance problems due to the high number of active concurrent SMB connections on each protocol node of IBM Spectrum Scale.

If IBM Spectrum Scale experiences performance problems, the following actions can be taken:

- You can use the IBM Spectrum Scale center GUI or CLI to understand which physical resource such as CPU, memory, networking, disk in the system are highly used resources. This information helps to gain an insight on the physical system resource that is possibly inhibiting or limiting performance. For more information, see the *mmperfmon* command topic in the *IBM Spectrum Scale: Command and Programming Reference*.
- Ensure that the IBM Spectrum Scale protocol nodes are configured with the maximum number of processors, memory, and networking adapters.
- Add more IBM Spectrum Scale protocol nodes to your system.
- Move certain advanced functions, such as TSM and AFM, to periods of time when SMB file activities that are related to the server is lower.
- Reduce the frequency at which snapshots are being created and deleted either at the file system or the file set level or both, especially during the periods of highest SMB user activity.
- Check that the file system cluster has appropriate values for pagepool, number of worker threads, and number of receiver threads. For more information, see the *Configuring and tuning your system for GPFS* topic in the *IBM Spectrum Scale: Administration Guide*.
- Investigate and fine-tune the performance of the underlying disk storage systems that contain the file systems on which the SMB export resides. The checklist items are described in the following list:
 - Ensure that the file system disks of a storage system are distributed between the pair of IBM Spectrum Scale storage nodes to which that storage system is attached. One half of the file system disks in a storage system should have one of the storage nodes that are identified as the primary NSD server. The other half of the file system disks in the storage system should have the other IBM Spectrum Scale storage node in the pair that is assigned as the primary NSD server. The **mmldisk** CLI command option shows the IBM Spectrum Scale storage nodes that are the primary and secondary NSD server for each file system disk.
 - If an IBM Spectrum Scale metadata replication or data replication is being used, ensure that you assign file system disks to failure groups that balance the I/O and data across a set of file system disks, RAID arrays, and disk storage systems. The **mmldisk** CLI command shows the failure group to which each file system disk is assigned. The **mmchdisk** CLI command can be used to change the failure group to which each file system disk is assigned.
 - If the underlying disk storage systems on which the file system reside is becoming a performance bottleneck, consider adding more physical resources to those disk storage systems. More resources include more cache memory, more disk drives, more RAID arrays and more file system disks for the file systems that contain the SMB exports.

- If the existing disk storage systems on which the file system resides reach their limit in terms of either capacity or performance or both, then consider adding more disk storage systems and extending the file system on which the SMB exports resides. Capacity and performance improvement can be done by adding new file system disks that are residing on the new disk storage systems to the existing disc storage system.

Considerations for SMB clients

Recommendations to consider before you mount SMB share on Linux systems.

Linux systems can mount SMB shares from protocol nodes. For more information, see your Linux distribution documentation and the `mount.cifs` man page. Workloads running on Linux SMB clients that often query `stat()` metadata can run into performance limitations. The performance limitation is due to the increase in processor usage as a result of the high number of metadata queries in the SMB protocol and the IBM Spectrum Scale SMB implementation.

The Linux SMB mount offers the "actimeo" mount option (see "mount.cifs" man page) for metadata caching on the Linux SMB client. The "actimeo" mount option reduces the metadata queries to the IBM Spectrum Scale SMB server. For workloads with a high number of metadata queries, you can set this parameter to improve the overall performance.

Fileset considerations for creating protocol data exports

You can create exports on the entire file system, on sub-directories of a file system, or on filesets.

A fileset is a file system object that enables you to manage data at a finer granularity than the file system. You can perform administrative operations such as defining quotas, creating snapshots, and defining file placement policies and rules, and specifying inode space values at the fileset level, especially when the fileset is independent.

In IBM Spectrum Scale, you can create exports even without filesets. Depending on your data management strategy, choose either of the following ways to create exports:

Create exports on the entire file system or on sub-directories of the file system

In this option, the export represents a large space. You can create independent filesets over the directories in this space and have finer control over the export directory paths. Universities and organizations that require a departmental multi-tenancy solution can choose this option.

Review the following example to better understand this option.

As a storage administrator of an organization, you want to create separate storage space for every department and user of the organization:

1. Export the root directory of the file system.

```
mmnfs export add /gpfs/fs0
```

Note: You can create a sub-directory in the root directory and export it. For example: `mmnfs export add /gpfs/fs0/home`

For more information, see **`mmnfs` command** in *IBM Spectrum Scale: Command and Programming Reference*.

2. Create independent filesets in the root directory, linked to the subdirectory `/gpfs/fs0/home`.

In the following example, it is assumed that there is a user `user1` that is a part of the group `group/HR`.

```
mmcrfileset fs0 hr_fileset --inode-space=new
mmlinkfileset fs0 hr_fileset -J /gpfs/fs0/home/hr
mmcrfileset fs0 user1_fileset --inode-space=new
mmlinkfileset fs0 user1_fileset -J /gpfs/fs0/home/user1
```

For more information, see the following commands in the *IBM Spectrum Scale: Command and Programming Reference*.

- **mmcrfileset command**
- **mmlinkfileset command**

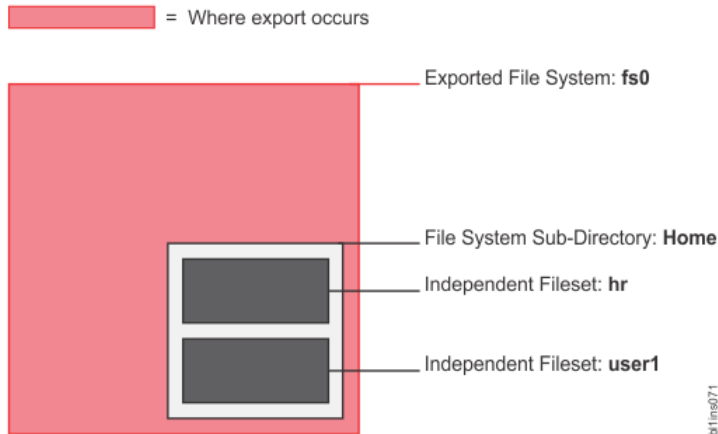
3. Similarly, create independent filesets for other departments and users.

You can assign quota to each user and department via the independent filesets.

The NFS and SMB clients can now mount the export and access their directories.

Access to the data in the export directories is controlled via the group and user ACLs. In this case, only the users who are in group HR, which has group ACLs to read/write into the hr directory, can access the directory. The user user1 who is in the group group/HR can perform read/write to the user1 directory and the hr directory.

Create exports on the entire file system or on sub-directories of the file system



Create exports on independent filesets

In this option, the independent filesets represent discrete projects. One or more exports can be created on each fileset. You can apply the Information Lifecycle Management (ILM) policies over the filesets to automate the placement and management of file data. You can protect the export data by granting access permissions to specific workstations or IP addresses. Also, data in the exports can be preserved by the independent fileset's snapshot policy.

Review the following example to better understand this option.

You are a storage administrator of a private cloud hosting storage system that stores webcam data. You want to ensure that a webcam has access only to its storage directory. Also, you want the data analyzers to access all data so that they can look for activity and generate analytical reports.

1. Create an independent fileset web_cam_data.

```
mmcrfileset fs0 web_cam_data --inode-space=new
mmlinkfileset fs0 web_cam_data -J /gpfs/fs0/web_cam_data
```

Data from all webcams is stored in /gpfs/fs0/web_cam_data.


2. Create exports for both webcams.

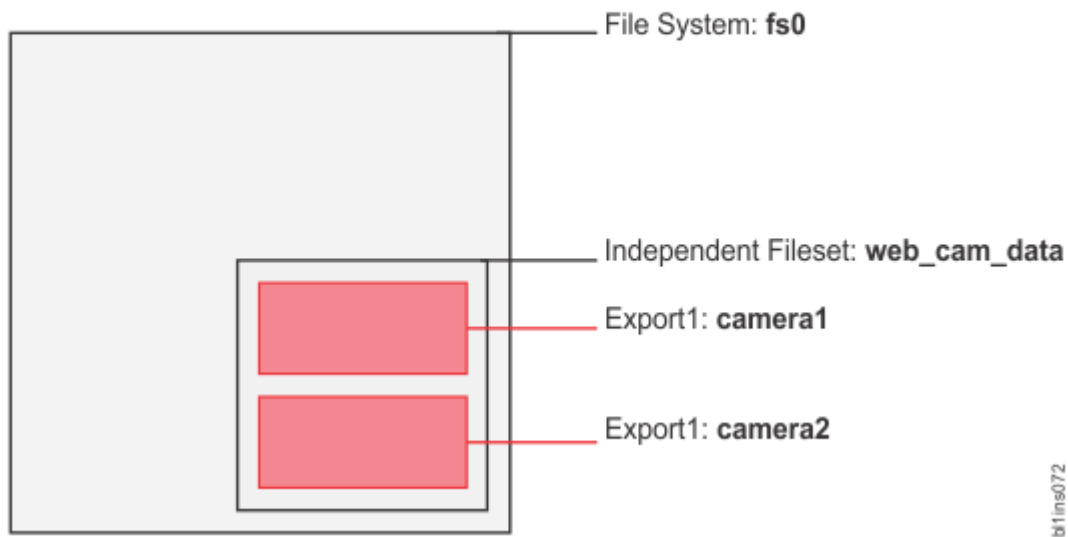
```
mkdir /gpfs/fs0/web_cam_data/camera1
mkdir /gpfs/fs0/web_cam_data/camera2

mmnfs export add "/gpfs/fs0/web_cam_data/camera1" \
-c "198.51.100.2(Access_Type=RW);203.0.113.2(Access_Type=R0);203.0.113.3(Access_Type=R0)"
mmnfs export add "/gpfs/fs0/web_cam_data/camera2" \
-c "198.51.100.3(Access_Type=RW);203.0.113.2(Access_Type=R0);203.0.113.3(Access_Type=R0)"
```

The webcam1 (IP: 198.51.100.2) mounts and records data to the camera1 export and the webcam2 (IP: 198.51.100.3) mounts and records data to the camera2 export. The data analyzers (IP: 203.0.113.2 and 203.0.113.3) are given 'Read Only' type access to both exports. Thus, the data is accessible only from the specified IP addresses.

Create exports on independent filesets

 = Where export occurs



Planning for Object Storage deployment

There are a number of decisions that must be made before you begin the Object Storage deployment in an IBM Spectrum Scale environment. The following information is an outline of these decision points.

The IBM Spectrum Scale for Object Storage architecture is shown in [Figure 39 on page 318](#).

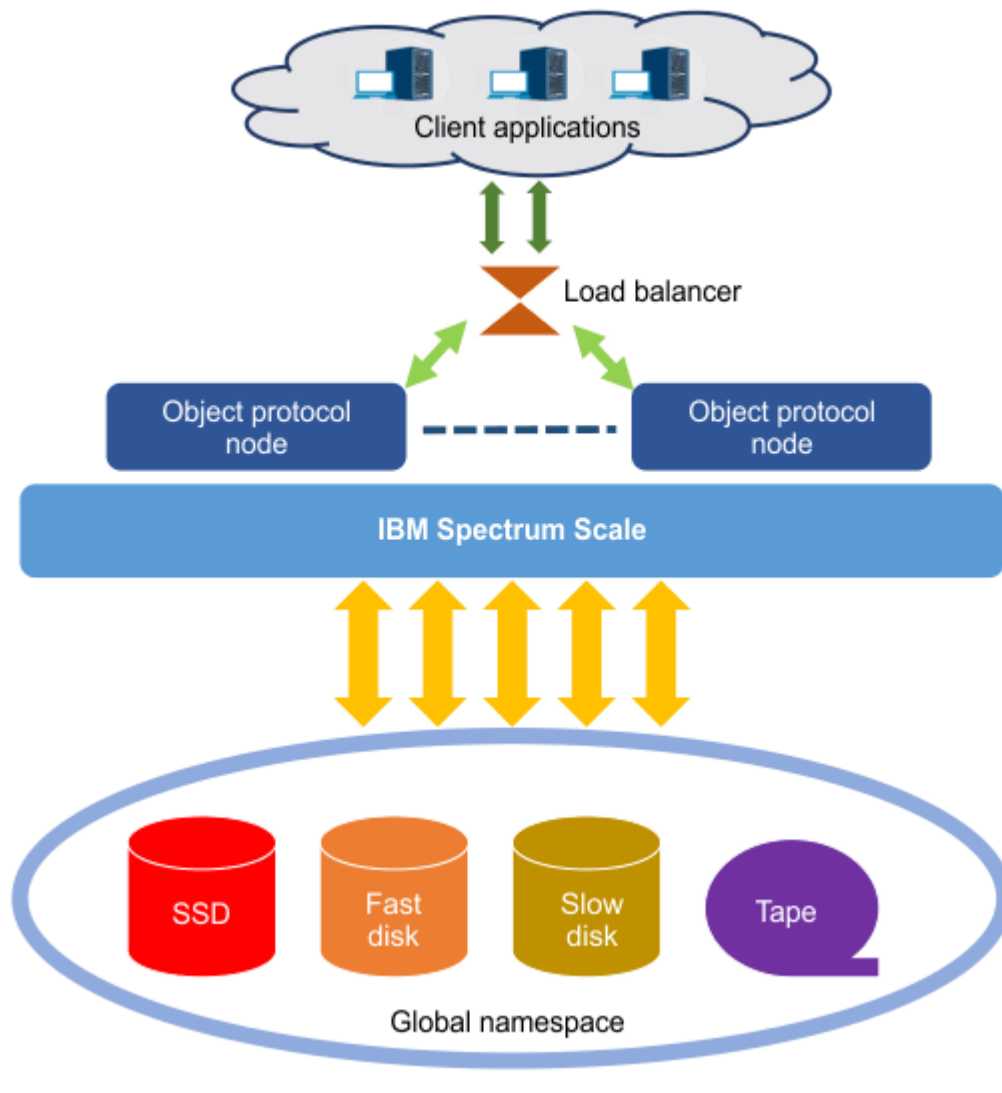


Figure 39. IBM Spectrum Scale for Object Storage architecture

For every object PUT request, the following inodes are created:

- A hashes directory for a PUT request of every new object is created.
- A target object file is created.
- The parent directory of the hash directory if it does not exist is created.
- The partition directory unless it exists is created.
- A hashes .pk1 file is created.
- A temporary .lock file is created.

Each protocol node runs all OpenStack Swift object services, the Keystone identity service, and the IBM Spectrum Scale client. Client applications make requests to complete object operations, such as uploading an object, downloading an object, and deleting an object.

The request is routed to a protocol node typically by a load balancer or by using DNS round robin. The protocol node implements that request by creating, retrieving, or deleting the object on the backend IBM Spectrum Scale storage. The request completion status is then returned to the client application. Each client request must include an authentication token. Typically, client applications request a token from the Keystone service and then provide that token in the subsequent object requests - until the token expires. At that point, a new token can be requested.

Client applications make requests to complete operations such as uploading an object, downloading an object, and deleting an object. Account and container information can also be updated, viewed, and removed by client applications. Use the OpenStack Swift API documentation available to clients. For more information about using the OpenStack Swift API, see <http://developer.openstack.org/api-ref/object-storage/index.html>.

For more information on OpenStack Swift, see [OpenStack Swift documentation](#).

For more information about using OpenStack Keystone with Swift, see [Keystone Auth section in the OpenStack Swift documentation](#).

OpenStack repository configuration required by the object protocol

On IBM Spectrum Scale 5.1.2.0 and earlier 5.1.x releases, an installation repository for the OpenStack packages and their dependencies must be configured on all protocol nodes before you install the object protocol.

Note: This repository configuration is not required in IBM Spectrum Scale 5.1.2.1 and later releases.

The installation of the object protocol requires an OpenStack repository to be preconfigured on all protocol nodes. This repository is needed to provide the Swift and Keystone OpenStack components, and their associated dependent packages. This repository can be configured through the Red Hat subscription manager or from publicly available OpenStack repositories.

Configuring the OpenStack repository from the Red Hat subscription manager

Red Hat supplies the OpenStack installation repository through the [Red Hat OpenStack Platform subscription](#). Contact Red Hat to add this subscription to your Red Hat license if necessary.

1. Attach the OpenStack pool ID to your subscription manager with the following command.

```
# subscription-manager attach --pool=12345...
```

You can obtain the OpenStack pool ID that is needed for this step by using the following command.

```
# subscription-manager list --all --available
...
  Red Hat OpenStack for IBM Power
  Red Hat Enterprise Linux for Real Time for NFV
...
SKU:          SER999
Contract:     112233
Pool ID:     12345...
```

2. Add the following repositories to the subscription manager. In the repository names, replace *ARCH* with *ppc64le* or *x86_64*, as needed for your environment.

```
# subscription-manager repos --enable=openstack-16-for-rhel-8-ARCH-rpms
# subscription-manager repos --enable=codeready-builder-for-rhel-8-ARCH-rpms
```

3. Repeat these steps on each protocol node to ensure that the OpenStack repositories are available on all of them.

Configuring the OpenStack repository from publicly available repositories

You can configure the necessary OpenStack packages from public providers. Refer to [OpenStack packages for RHEL and CentOS in Open Stack documentation](#) for instructions on how to use CentOS or RDO repositories. Ensure that the Train version of the OpenStack repository is configured.

Verifying that the OpenStack repository is properly configured

Ensure that the protocol nodes can access the OpenStack repository and packages with the following command.

```
# dnf whatprovides python3-keystone
```

```
python3-keystone-1:16.0.1-0.20191210095025.bd3f637.el8ost.noarch : Keystone Python libraries  
Repo      : openstack-16-for-rhel-8-ppc64le-rpms  
Matched from:  
Provide   : python3-keystone = 1:16.0.1-0.20191210095025.bd3f637.el8os
```

Related concepts

[Load balancing](#)

Use multiple protocol nodes to provide higher throughput and performance of Object Storage by allowing more requests in parallel and distributing the workload among all protocol nodes.

[Cluster host name](#)

The cluster host name is required during the installation process.

[Authentication method](#)

IBM Spectrum Scale for object storage supports a flexible array of configuration options for authentication.

[Backup and disaster recovery strategy](#)

[SELinux considerations](#)

To simplify the configuration of the IBM Spectrum Scale for Object Storage environment, the installation process detects whether SELinux is enabled or not. If SELinux is enabled, the installation process performs steps so that the object services and the database software that runs on the protocol nodes can interact with the required file system and system resources.

[Eventual consistency model](#)

An eventual consistency model is used when you upload or delete object data.

[Planning for unified file and object access](#)

You can use unified file and object access to access data by using object and file interfaces. Before you use unified file and object access, you must plan for a number of aspects.

[Planning for multi-region object deployment](#)

Use the following information to plan your multi-region object deployment.

Load balancing

Use multiple protocol nodes to provide higher throughput and performance of Object Storage by allowing more requests in parallel and distributing the workload among all protocol nodes.

Certain steps might be necessary to ensure that client requests are distributed among all protocol nodes. Object store endpoint URLs stored in keystone contain the destination host name or IP addresses. Therefore, it is important to ensure that requests that are sent to that destination are distributed among all the protocol addresses. This endpoint address is the value that is specified by the **--endpoint** parameter when you set up Object Storage with the installation toolkit.

If a host name is resolved to a single protocol IP address (such as 192.168.1.1), then all client requests are sent to the protocol node associated with that specific address. Make sure that requests are distributed among all the protocol nodes instead of just a single protocol node.

To accomplish this you can use a load balancer, such as HAProxy. In this case, the destination for the endpoint is configured to resolve to the load balancer, and the load balancer forwards incoming requests to one of the protocol nodes (based on a balancing strategy). By configuring the load balancer with all of the protocol IP addresses, the client requests can be sent to the entire set of protocol nodes.

Another common solution is to use a DNS Round Robin. In this case, a DNS server is configured so a single host name is associated with a set of addresses. When a client does a DNS lookup of the host name, one address from the set is returned. By configuring the DNS Round Robin service to associate the set of protocol IP addresses with the endpoint host name, client requests are distributed among all protocol nodes. As each client initiates a request to the endpoint hostname, the DNS Round Robin service returns an IP address by cycling through the set of protocol IP addresses.

Related concepts

[OpenStack repository configuration required by the object protocol](#)

Cluster host name

The cluster host name is required during the installation process.

Authentication method

IBM Spectrum Scale for object storage supports a flexible array of configuration options for authentication.

Backup and disaster recovery strategy

SELinux considerations

To simplify the configuration of the IBM Spectrum Scale for Object Storage environment, the installation process detects whether SELinux is enabled or not. If SELinux is enabled, the installation process performs steps so that the object services and the database software that runs on the protocol nodes can interact with the required file system and system resources.

Eventual consistency model

An eventual consistency model is used when you upload or delete object data.

Planning for unified file and object access

You can use unified file and object access to access data by using object and file interfaces. Before you use unified file and object access, you must plan for a number of aspects.

Planning for multi-region object deployment

Use the following information to plan your multi-region object deployment.

Cluster host name

The cluster host name is required during the installation process.

It is the **endpoint** parameter in the **spectrumscale config object** command and it is the **cluster-hostname** parameter in the **mmobj swift base** command. For more information, see *mmobj command* and *spectrumscale command* in *IBM Spectrum Scale: Command and Programming Reference*. Additionally, the cluster host name might be used in your load balancer or DNS round robin configuration. It is the fully qualified name that clients send their object requests to.

Related concepts

OpenStack repository configuration required by the object protocol

Load balancing

Use multiple protocol nodes to provide higher throughput and performance of Object Storage by allowing more requests in parallel and distributing the workload among all protocol nodes.

Authentication method

IBM Spectrum Scale for object storage supports a flexible array of configuration options for authentication.

Backup and disaster recovery strategy

SELinux considerations

To simplify the configuration of the IBM Spectrum Scale for Object Storage environment, the installation process detects whether SELinux is enabled or not. If SELinux is enabled, the installation process performs steps so that the object services and the database software that runs on the protocol nodes can interact with the required file system and system resources.

Eventual consistency model

An eventual consistency model is used when you upload or delete object data.

Planning for unified file and object access

You can use unified file and object access to access data by using object and file interfaces. Before you use unified file and object access, you must plan for a number of aspects.

Planning for multi-region object deployment

Use the following information to plan your multi-region object deployment.

Authentication method

IBM Spectrum Scale for object storage supports a flexible array of configuration options for authentication.

If you already have Keystone that is deployed in your environment, you can configure IBM Spectrum Scale for object storage to use the existing or external Keystone. If you configure Keystone as a part a IBM Spectrum Scale cluster, you can manage the user information locally, or you can integrate it with a Microsoft Active Directory or LDAP system.

Related concepts

[OpenStack repository configuration required by the object protocol](#)

[Load balancing](#)

Use multiple protocol nodes to provide higher throughput and performance of Object Storage by allowing more requests in parallel and distributing the workload among all protocol nodes.

[Cluster host name](#)

The cluster host name is required during the installation process.

[Backup and disaster recovery strategy](#)

[SELinux considerations](#)

To simplify the configuration of the IBM Spectrum Scale for Object Storage environment, the installation process detects whether SELinux is enabled or not. If SELinux is enabled, the installation process performs steps so that the object services and the database software that runs on the protocol nodes can interact with the required file system and system resources.

[Eventual consistency model](#)

An eventual consistency model is used when you upload or delete object data.

[Planning for unified file and object access](#)

You can use unified file and object access to access data by using object and file interfaces. Before you use unified file and object access, you must plan for a number of aspects.

[Planning for multi-region object deployment](#)

Use the following information to plan your multi-region object deployment.

Backup and disaster recovery strategy

For information on backup and disaster recovery strategy for your Object Storage, see the *Protocols cluster disaster recovery* section in the *IBM Spectrum Scale: Administration Guide*.

Related concepts

[OpenStack repository configuration required by the object protocol](#)

[Load balancing](#)

Use multiple protocol nodes to provide higher throughput and performance of Object Storage by allowing more requests in parallel and distributing the workload among all protocol nodes.

[Cluster host name](#)

The cluster host name is required during the installation process.

[Authentication method](#)

IBM Spectrum Scale for object storage supports a flexible array of configuration options for authentication.

[SELinux considerations](#)

To simplify the configuration of the IBM Spectrum Scale for Object Storage environment, the installation process detects whether SELinux is enabled or not. If SELinux is enabled, the installation process performs steps so that the object services and the database software that runs on the protocol nodes can interact with the required file system and system resources.

[Eventual consistency model](#)

An eventual consistency model is used when you upload or delete object data.

Planning for unified file and object access

You can use unified file and object access to access data by using object and file interfaces. Before you use unified file and object access, you must plan for a number of aspects.

Planning for multi-region object deployment

Use the following information to plan your multi-region object deployment.

SELinux considerations

To simplify the configuration of the IBM Spectrum Scale for Object Storage environment, the installation process detects whether SELinux is enabled or not. If SELinux is enabled, the installation process performs steps so that the object services and the database software that runs on the protocol nodes can interact with the required file system and system resources.

The `openstack-selinux` package is installed automatically when the `spectrum-scale-object` package is installed. This packages installation configures the object services for SELinux.

If the installer detects that SELinux is enabled, it does the following steps:

1. Ensures that the Postgres database can access the Keystone database directory on the CES shared root file system:

```
semanage fcontext -a -t postgresql_db_t "<keystone db directory>(/.*)?"
semanage fcontext -a -t postgresql_log_t "<keystone db directory>/
log(/.*)?"
restorecon -R "<keystone db directory>"
```

2. Ensures that object processes can access the object data fileset:

```
semanage fcontext -a -t swift_data_t "<object fileset directory>(/.*)?"
restorecon -R <object fileset directory>/*
```



Attention:

- The object protocol is not supported in IBM Spectrum Scale 5.1.0.0. If you want to deploy object, install the IBM Spectrum Scale 5.1.0.1 or a later release.
- If SELinux is disabled during installation of IBM Spectrum Scale for object storage, enabling SELinux after installation is not supported.

SELinux packages required for IBM Spectrum Scale for Object Storage

When the IBM Spectrum Scale object protocol is installed, the following SELinux packages are also installed:

- `selinux-policy-base` at 3.13.1-23 or higher
- `selinux-policy-targeted` at 3.12.1-153 or higher

When you use the object protocol, you cannot enable SELinux after the IBM Spectrum Scale installation. Contact IBM Spectrum Scale support by sending an email to scale@us.ibm.com, if you have questions about this restriction.

Related concepts

[OpenStack repository configuration required by the object protocol](#)

[Load balancing](#)

Use multiple protocol nodes to provide higher throughput and performance of Object Storage by allowing more requests in parallel and distributing the workload among all protocol nodes.

[Cluster host name](#)

The cluster host name is required during the installation process.

[Authentication method](#)

IBM Spectrum Scale for object storage supports a flexible array of configuration options for authentication.

[Backup and disaster recovery strategy](#)

[Eventual consistency model](#)

An eventual consistency model is used when you upload or delete object data.

[Planning for unified file and object access](#)

You can use unified file and object access to access data by using object and file interfaces. Before you use unified file and object access, you must plan for a number of aspects.

[Planning for multi-region object deployment](#)

Use the following information to plan your multi-region object deployment.

Eventual consistency model

An eventual consistency model is used when you upload or delete object data.

According to the CAP theorem, in distributed storage systems, a system can guarantee two of the following three claims:

1. Consistency
2. Availability
3. Partition tolerance

With Swift and IBM Spectrum Scale for Object Storage, the availability and partition tolerance claims are chosen. Therefore, in some cases, it is possible to get an inconsistent listing of object store contents. The most common case is consistency between container listing databases and objects on disk. If a normal object upload, the object data is committed to the storage and the container listing database is updated. Under heavy load, it is possible that the container listing database update does not complete immediately. In that case, the update is made asynchronously by the object-updater service. In the time between the original object commit and when the object-updater executes, you do not see the new object in the container listing, even though the object is safely committed to storage. A similar situation can occur with object deletions.

Related concepts

[OpenStack repository configuration required by the object protocol](#)

[Load balancing](#)

Use multiple protocol nodes to provide higher throughput and performance of Object Storage by allowing more requests in parallel and distributing the workload among all protocol nodes.

[Cluster host name](#)

The cluster host name is required during the installation process.

[Authentication method](#)

IBM Spectrum Scale for object storage supports a flexible array of configuration options for authentication.

[Backup and disaster recovery strategy](#)

[SELinux considerations](#)

To simplify the configuration of the IBM Spectrum Scale for Object Storage environment, the installation process detects whether SELinux is enabled or not. If SELinux is enabled, the installation process performs steps so that the object services and the database software that runs on the protocol nodes can interact with the required file system and system resources.

[Planning for unified file and object access](#)

You can use unified file and object access to access data by using object and file interfaces. Before you use unified file and object access, you must plan for a number of aspects.

[Planning for multi-region object deployment](#)

Use the following information to plan your multi-region object deployment.

Planning for unified file and object access

You can use unified file and object access to access data by using object and file interfaces. Before you use unified file and object access, you must plan for a number of aspects.

Related concepts

[OpenStack repository configuration required by the object protocol](#)

[Load balancing](#)

Use multiple protocol nodes to provide higher throughput and performance of Object Storage by allowing more requests in parallel and distributing the workload among all protocol nodes.

[Cluster host name](#)

The cluster host name is required during the installation process.

[Authentication method](#)

IBM Spectrum Scale for object storage supports a flexible array of configuration options for authentication.

[Backup and disaster recovery strategy](#)

[SELinux considerations](#)

To simplify the configuration of the IBM Spectrum Scale for Object Storage environment, the installation process detects whether SELinux is enabled or not. If SELinux is enabled, the installation process performs steps so that the object services and the database software that runs on the protocol nodes can interact with the required file system and system resources.

[Eventual consistency model](#)

An eventual consistency model is used when you upload or delete object data.

[Planning for multi-region object deployment](#)

Use the following information to plan your multi-region object deployment.

Planning for identity management modes for unified file and object access

As you plan for unified file and object access, it is important to understand the identity management modes for unified file and object access.

Based on the identity management mode that you plan to use, you must plan for the authentication mechanism to be configured with file and object. In an existing IBM Spectrum Scale setup in which an authentication mechanism is already set up, a suitable identity management mode must be chosen for unified file and object access.

For more information, see *Identity management modes for unified file and object access* in *IBM Spectrum Scale: Administration Guide*.

Authentication planning for unified file and object access

Planning to use a suitable authentication mechanism for a unified file and object access setup requires an understanding of the identity management modes for unified file and object access, and the authentication setups that are supported by these modes.

For more information, see *Authentication in unified file and object access* and *Identity management modes for unified file and object access* in *IBM Spectrum Scale: Administration Guide*.

ibmobjectizer service schedule planning

Use the unified file and object access feature to access data that is ingested through the unified file to be accessed from object.

This access is enabled by a process that is called objectization, which is done by the **ibmobjectizer** service. The **ibmobjectizer** service runs periodically and converts the file data that is located under unified file and object access enabled filesets to make it available for object access.

It is important to understand the amount and frequency of file data that is expected to be ingested and objectized because the objectization service must be scheduled by the administrator.

For information about the **ibmobjectizer** service, see *The objectizer process, Setting up the objectizer service interval, and Configuration files for IBM Spectrum Scale for object storage* in *IBM Spectrum Scale: Administration Guide*.

Prerequisites for unified file and object access

To enable unified file and object access, you must enable the file-access object capability.

For more information, see [“Object capabilities” on page 37](#).

For unified file and object access, the authentication prerequisites depend on the identity management mode for unified file and object access that you plan to use. It is important to choose the mode before you configure unified file and object access, although you can move from one mode to another. For more information about changing identity management modes for unified file and object access, see *Configuring authentication and setting identity management modes for unified file and object access* in *IBM Spectrum Scale: Administration Guide*.

For more information about authentication and identity management, see *Authentication in unified file and object access* and *Identity management modes for unified file and object access* in *IBM Spectrum Scale: Administration Guide*.

Unified file and object access are deployed as a storage policy for object storage. Therefore, it is important to understand the concept of a storage policy for object storage and how to associate containers with storage policies. For more information, see [“Storage policies for Object Storage” on page 34](#).

Planning for multi-region object deployment

Use the following information to plan your multi-region object deployment.

Enabling multi-region enables the primary rings to be multi-region and all data to be stored in all regions. Storage policies can be used to limit objects to certain regions.

For information about enabling multi-region object deployment, see [“Enabling multi-region object deployment initially” on page 448](#).

For information on adding a region to a multi-region object deployment environment, see *Adding a region in a multi-region object deployment* in *IBM Spectrum Scale: Administration Guide*.

Related concepts

[OpenStack repository configuration required by the object protocol](#)

[Load balancing](#)

Use multiple protocol nodes to provide higher throughput and performance of Object Storage by allowing more requests in parallel and distributing the workload among all protocol nodes.

[Cluster host name](#)

The cluster host name is required during the installation process.

[Authentication method](#)

IBM Spectrum Scale for object storage supports a flexible array of configuration options for authentication.

[Backup and disaster recovery strategy](#)

[SELinux considerations](#)

To simplify the configuration of the IBM Spectrum Scale for Object Storage environment, the installation process detects whether SELinux is enabled or not. If SELinux is enabled, the installation process performs steps so that the object services and the database software that runs on the protocol nodes can interact with the required file system and system resources.

[Eventual consistency model](#)

An eventual consistency model is used when you upload or delete object data.

[Planning for unified file and object access](#)

You can use unified file and object access to access data by using object and file interfaces. Before you use unified file and object access, you must plan for a number of aspects.

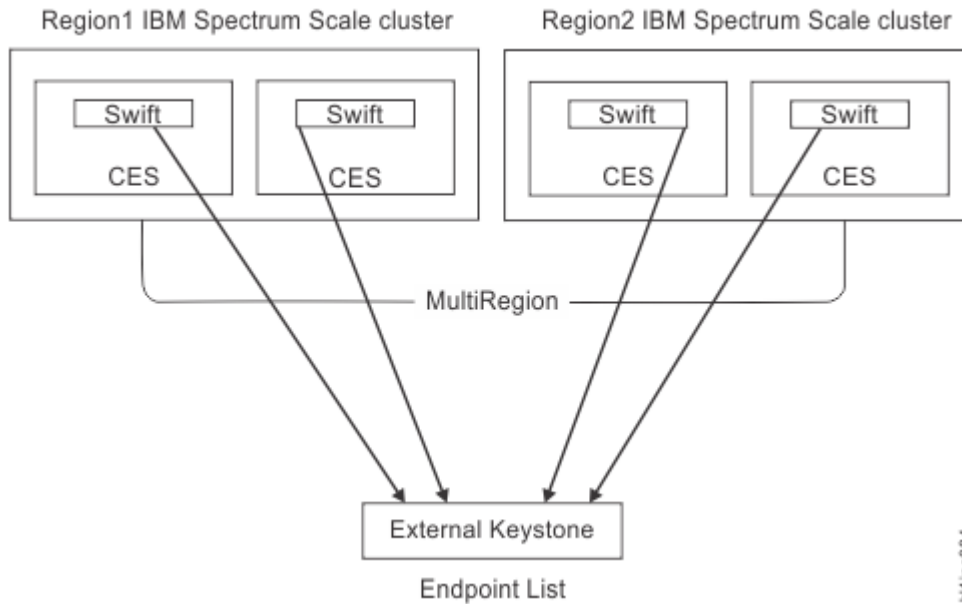
Authentication considerations for multi-region object deployment

In a multi-region object deployment environment, all regions must use the same Keystone service.

The Keystone service can be a local Keystone server that is installed with the object deployment or it can be an independent service. Subsequent clusters that join the environment must specify an external Keystone server during installation.

The following two methods can be used for object authentication configuration with a multi-region setup:

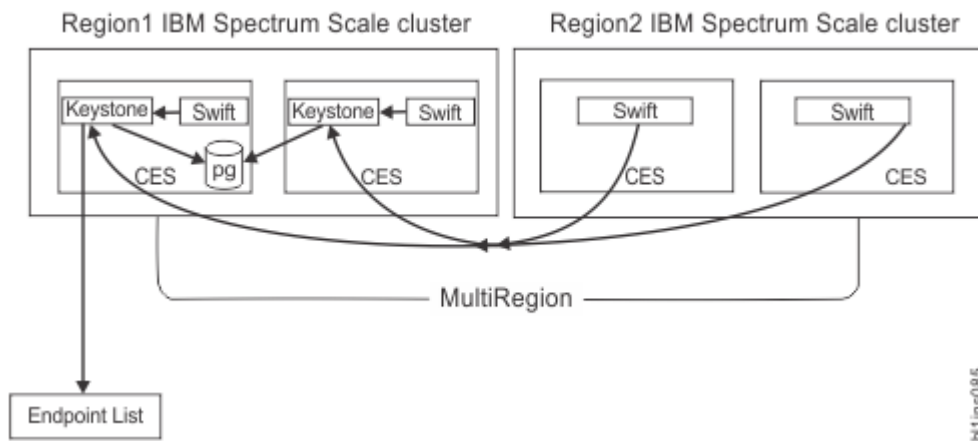
- You can use the external Keystone server for object authentication configuration.



ID	Region	Service Name	Service Type	Enabled	Interface	URL
e310	Region0ne	swift	object-store	True	public	http://region1:8080/v1/AUTH_%(tenant_id)s
1679	Region0ne	swift	object-store	True	internal	http://region1:8080/v1/AUTH_%(tenant_id)s
c458	Region0ne	swift	object-store	True	admin	http://region1:8080
8a01	Region2	swift	object-store	True	public	http://region2:8080/v1/AUTH_%(tenant_id)s
b821	Region2	swift	object-store	True	internal	http://region2:8080/v1/AUTH_%(tenant_id)s
5188	Region2	swift	object-store	True	admin	http://region2:8080

Important: The external keystone and HA must be managed and configured by the customer.

1. Configure the external keystone with the **mmobj** command on all the participant clusters of the multi-region setup.
 2. Use the **mmobj swift base** command with the **--remote-keystone-url** and **--configure-remote-keystone** arguments.
- You can use the keystone server that is installed on one of the IBM Spectrum Scale clusters.



ID	Region	Service Name	Service Type	Enabled	Interface	URL
e310	RegionOne	swift	object-store	True	public	http://region1:8080/v1/AUTH_%(tenant_id)s
1679	RegionOne	swift	object-store	True	internal	http://region1:8080/v1/AUTH_%(tenant_id)s
c458	RegionOne	swift	object-store	True	admin	http://region1:8080
8a01	Region2	swift	object-store	True	public	http://region2:8080/v1/AUTH_%(tenant_id)s
b821	Region2	swift	object-store	True	internal	http://region2:8080/v1/AUTH_%(tenant_id)s
5188	Region2	swift	object-store	True	admin	http://region2:8080

Note: If the region1 cluster stops functioning, the complete multi-region setup is unusable because the keystone service is not available.

1. On the first cluster of multi-region setup, configure local Keystone with the **mmobj** command by using the installation toolkit.
2. Use the **mmobj swift base** command with the **--local-keystone** arguments for configuring with keystone with local authentication type.
3. For configuring the object authentication with **ad | ldap**, use the **mmuserauth service create|delete** command after **mmobj swift base** with **--local-keystone**.
4. On the second and third clusters of the multi-region setup, configure the external keystone with the **mmobj** command.
5. Use the **mmobj swift base** command with the **--remote-keystone-url** and **--configure-remote-keystone** arguments.

Data protection considerations for multi-region object deployment

There are data protection considerations for multi-region object deployment.

Each cluster must maintain appropriate backups. A multi-region cluster can be more resilient to failures because if a cluster loses multi-region objects or a single cluster fails, the lost objects are restored through the normal Swift replication behavior.

Ensure that the Keystone data is backed up because all regions use the same Keystone service.

Network considerations for multi-region object deployment

To communicate with nodes in other regions, the Swift services connect to the network addresses as defined in the ring files (which are the CES IP addresses).

Note: Every node must be able to connect to all CES IP addresses in all regions.

Make sure that the network routing is set up to enable the connections. Make sure that the necessary firewall configuration is set up to allow connection to the object, account, and container servers (typically ports 6200, 6201, and 6202) on all nodes in all regions. Swift uses the **rsync** command to replicate data between regions, so the **rsync** port 873 also must be opened between the regions.

Monitoring and callbacks in a multi-region object deployment setup

Within a region, monitoring multi-region ring files and distribution to nodes is dependent upon the existing monitoring framework and the work that is done to enable storage policies in the monitoring layer.

For the distribution of ring and configuration changes to other regions, the administrator must perform those steps manually by using the **mmobj multiregion** command. For more information, see *mmobj command* in *IBM Spectrum Scale: Command and Programming Reference*.

Performance considerations for multi-region object deployment

Before objects are replicated from one cluster to another, there is a delay because of the replicator run time and the network load.

Each region's access to the Keystone server affects the performance of authentication capabilities within its region.

Planning for CES HDFS

For CES HDFS planning information, see *Planning and Limitations and Recommendations* under [CES HDFS](#) in *Big data and analytics support documentation*.

Planning for Cloud services

Proper planning is one of the most important activities that you need to carry out for successful implementation of Cloud services on the IBM Spectrum Scale cluster.

Hardware requirements for Cloud services

You must meet certain hardware requirements to be able to install and use Cloud services on the IBM Spectrum Scale cluster.

For running Cloud services, adhere to the recommended CPU and memory requirements for a CES node even when installing Cloud services on an NSD node. For more information, see the [IBM Spectrum Scale FAQ](#) in *IBM Documentation*.

The hardware requirements for Cloud services are:

- Any standard x86 64-bit servers or Power Linux nodes that run supported Linux distributions.
- The minimum size that is required for the `/var/MCStore` folder is 12 GB.

Note: For better performance, it is recommended to have a minimum of 2 CPU socket server of the latest Intel variety with at least 128 GB of memory.

A high CPU count promotes better cloud tiering throughput because although object storage can be slow in I/O operations per thread, object storage can support many threads. Use sixteen or more CPUs when you select your hardware.

Cloud tiering services demand a large amount of memory, which is why the minimum recommended memory size is 128 GB. Memory size requirements increase if the number of files increases, as you add files on the cloud means you must increase the memory that is on your system. For larger deployments, it is recommended that you use 10 - 20 times as much memory that is required so the Cloud services can cache its directory database data.

Software requirements for Cloud services

You must meet certain software requirements to be able to install and use Cloud services on the IBM Spectrum Scale cluster.

Note: Cloud services are shipped along with IBM Spectrum Scale Advanced Edition, IBM Spectrum Scale Data Management Edition, or IBM Spectrum Scale Developer Edition or IBM Spectrum Scale Erasure Code Edition. Therefore, IBM Spectrum Scale Advanced Edition, IBM Spectrum Scale Data Management

Edition, or IBM Spectrum Scale Developer Edition or IBM Spectrum Scale Erasure Code Edition is required for the installation.

The software requirements for Cloud services are

- For the complete list of OS support, see [Table 32 on page 330](#).
- For Kernel versions, see [IBM Spectrum Scale FAQ in IBM Documentation](#).
- The following packages must be installed on the Transparent cloud tiering core server nodes:
 - Redhat-release-server
 - Unzip
 - Tar
 - bc - An arbitrary precision calculator language
 - SQLite 3.7 and above
 - GPFS base RPMs
 - GPFS RPMs required for a CES node
 - GPFS Advanced Edition RPMs
- For bulk delete functionality to work in Swift, verify that the Bulk middleware to be in pipeline, as follows:

```
vim /etc/swift/proxy-server.conf
[pipeline:main] pipeline = healthcheck cache bulk authtoken keystone proxy-server
```

- To run client-assist, clients need all the packages listed for server nodes except SQLite.

<i>Table 32. OS support matrix</i>		
OS	TCT server	TCT client
RHEL 7.1 (x86)	yes	yes
RHEL 7.2 (x86)	yes	yes
RHEL 7.3 (x86)	yes	yes
RHEL 7.4 (x86)	yes	yes
RHEL 7.5 (x86)	yes	yes
RHEL 7.1 (Power8 and Power9 LE)	yes	yes
RHEL 7.2 (Power8 and Power9 LE)	yes	yes
RHEL 7.3 (Power8 and Power9 LE)	yes	yes
RHEL 7.4 (Power8 and Power9 LE)	yes	yes
RHEL 7.5 (Power8 and Power9 LE)	yes	yes
RHEL 8.0 (Power8 and Power9 LE)	yes	yes
RHEL 7.5 (s390x)	yes	yes
SLES 11.2	no	yes
SLES 12.0	no	yes

Table 32. OS support matrix (continued)		
OS	TCT server	TCT client
SLES12 SP3 (s390x)	no	yes
Ubuntu 14.x and 16.4	no	yes
Ubuntu 16.04 and 18.04 (s390x)	no	yes

For the latest support matrix, see *Transparent cloud tiering questions* in [IBM Spectrum Scale FAQ](#).

Network considerations for Cloud services

This topic describes the network considerations that you need to follow for installing Cloud services on your IBM Spectrum Scale cluster.

Note:

- Network Time Protocol (NTP) must be correctly set on all nodes where the Cloud services are installed. If correct date and time are not set, some of the cloud storage providers will refuse to authenticate your requests.
- Cloud services always use the GPFS cluster (daemon) network and not the CES network.

Cloud storage supports storing cold data on a public cloud object storage service, such as IBM Softlayer or Amazon S3. When using a public cloud, the connection is over a WAN.

Cloud services for both Transparent cloud tiering and Cloud data Sharing can consume a considerable amount of WAN bandwidth. Customers need to ensure that there is sufficient WAN bandwidth for migrations and recalls along with any other WAN usage that they might require. When Cloud services run, an administrator can monitor WAN usage and other running processes in the IBM Spectrum Scale performance monitor. For a list of metrics, see the *Cloud services* section in the *List of Metrics* topic in the *IBM Spectrum Scale: Administration Guide*.

To avoid any conflict of Cloud services I/O traffic and internal IBM Spectrum Scale cluster I/O traffic, it is recommended that the Cloud services node is configured with two or more NICs. One NIC card is designated as the target for all inter-cluster I/O, and the other one is designated for Cloud services I/O traffic.

After the NIC card is set up and connected to the WAN, it is recommended to run any external network speed test, such as speedtest.net, to ensure that the I/O bandwidth to the external network meets your expectations.

If customers choose to configure a private cloud instance, they need to ensure that there is sufficient bandwidth between the IBM Spectrum Scale cluster and the private cloud to support migrating and recalling files. It is preferred to have a 10 GB network for higher throughput.

Cluster node considerations for Cloud services

This topic describes the cluster considerations that need to be followed before you install Cloud services on the IBM Spectrum Scale cluster.

You can install the core Cloud services packages on a maximum of 4 CES or NSD (or a combination of both) nodes that you want as members of your Cloud services node class. We support up to 4 node classes. The client package is installed on all other nodes within the same cluster. Cloud data transfer (migration, recall, import, or export) occurs only from the Cloud services nodes. However, you can initiate a cloud data transfer from either a server or a client node.

You can install the transparent cloud tiering server package on a non-gateway node. In this client-assisted recall configuration, the non-gateway node can recall from the same node instead of routing the commands through the gateway nodes. Except for recall, all other commands are routed through the gateway nodes. Thus, file-read requests originating on the client nodes are served immediately from the same node.

Use of client-assisted recall configuration is recommended in the following scenarios:

- The non-gateway node has direct network access to the cloud object storage.
- High CPU and memory resources are available on the node as the client-assist package starts a Java virtual machine per node-class managed
- If a large number of recalls are initiated through the non-gateway nodes.

Do not use a mixed node cluster configuration in which there are nodes on the cluster that do not allow the installation of clients. For example, Windows nodes do not allow the installation of clients because there is no Transparent cloud tiering Windows client. These configurations:

- Do not support transparent data recalling
- Cannot respond to requests that initiate from nodes on which a client is not installed
- Cannot efficiently perform deletions

Cloud services is supported for use only in IBM Spectrum Scale clusters and on any associated remotely mounted clients, with Intel x86 Linux and Power LE Linux nodes. Use of Windows, PowerPC Big Endian, Linux on Z or AIX nodes that access file systems, where Cloud services is used (either within the cluster or via a remotely mounted client), is untested and unsupported. Inclusion of such nodes is allowed only if there is no possibility that they will ever access Cloud services; for example, such nodes are tested and supported for PowerPC Big Endian nodes in Elastic Storage Server (ESS) BE & LE servers, where no user applications are supported on Elastic Storage Server (ESS).

IBM Cloud Object Storage considerations

The following information describes about the points that you need to consider before you use IBM Cloud Object Storage as the object storage provider.

Note: IBM Cloud Object Storage 3.7.3.2 or above is required for the Cloud services functions. If you are running an older version of IBM Cloud Object Storage, contact IBM support for upgrading your version.

Before you begin, ensure that an access pool is created and available on the IBM Cloud Object Storage system. The access pool must have 'Cloud Storage Object' configured as the API type.

Before you configure Cloud services with IBM Cloud Object Storage as the object storage provider, ensure that the following settings are done through IBM Cloud Object Storage **dsNet Manager** GUI. If these settings are not correctly set, the cloud account configuration or data migration fail.

- During the Cloud services setup process, two vaults are created on the IBM Cloud Object Storage system. One of the vaults is used for storing data and the other one is used for storing metadata. The data vault that is created by Cloud services contains the “container-prefix” appended with a unique ID.

Note: IBM Cloud Object Storage endpoint URL (“cloud_url” command-line option) must not have this container or vault name.

- In order for Cloud services to be able to create the vault, the user (whose access key is specified in the **mmcloudgateway account create** configuration command) must have the "Vault Provisioner" role that is assigned through the **dsNet Manager** UI.

To create vaults, go to **dsNet Manager** UI > **Security** tab and click **user > Roles > Add "Vault Provisioner"** role.

Make sure that either “Create Only” or “Create and Delete” permission is selected under **Administration > Provisioning API configuration**. Doing this enables Cloud services to create vaults by using the IBM Cloud Object Storage vault provisioning APIs. It is not necessary to allow Cloud services privileges to create the vault. You can create the vault separately by using IBM Cloud Object Storage management services.

Note: Delete access is not required for the “Vault Provisioner”.

Note: To specify the container name of an existing container, use the **--data-name/--meta-container** parameter of the **mmcloudgateway** command.

- For IBM Cloud Object Storage deployed on public cloud and with container mode enabled, you must contact the cloud object storage admin to obtain the accessor IP address or hostname, S3 credentials, and provisioning code (location) for the container vault.
- To create vaults through the provisioning API, IBM Cloud Object Storage uses provisioning templates. You can provide a provisioning template to the Cloud services in two ways:

Using default template

A default vault template can be set on the **dsNet Manager** UI. Go to **dsNet Manager** UI > **Configure** tab > **Storage pools** > **Create Vault template**. Then, **dsNet system** > **Default Vault template** and then select the newly created template. The vault that is created by Cloud services uses the default template.

Note: The default template on the IBM Cloud Object Storage system must have index that is enabled.

Using Provisioning code

If the default vault template is not set or not preferred, then a provisioning code can be used. Ensure that during the creation of the vault template, a unique user-defined provisioning code is specified. The provisioning code is different from the name.

To look up the provisioning code, go to **dsNet Manager** UI > **Configure** tab > **Storage pools** > **Vault Templates** > select the template, and look for “Provisioning Code”). Use the `--location` option in the **mmcloudgateway account create** command to specify the provisioning code. Using this mechanism, the vault template that is configured for Cloud services is used for vault provisioning.

Note: If there is no default provisioning template set on the IBM Cloud Object Storage system and a provisioning code is not specified to the **mmcloudgateway account create** command, the command fails. If the provisioning code specified is not found on the IBM Cloud Object Storage system, the command fails.

The following settings are recommended when you create a vault template on IBM Cloud Object Storage dedicated for Transparent cloud tiering.

Table 33. Recommended settings when you create a vault template on IBM Cloud Object Storage		
Configuration	Recommended Values	Comment
Width	See IBM Cloud Object Storage documented configuration for production.	
Threshold	See IBM Cloud Object Storage documented configuration for production.	
WriteThreshold	See IBM Cloud Object Storage documented configuration for production.	
Alert Level	See IBM Cloud Object Storage documented configuration for production.	
Alert Level	See IBM Cloud Object Storage documented configuration for production.	

Table 33. Recommended settings when you create a vault template on IBM Cloud Object Storage
(continued)

Configuration	Recommended Values	Comment
SecureSlice Technology	Disabled	When using Cloud data sharing services, the user might consider enabling SecureSlice Technology encryption. If using the Transparent cloud tiering services, encryption is not needed and is redundant since data is encrypted by the Transparent cloud tiering service before the data is stored on object storage.
SecureSliceAlgorithm	Not applicable since SecureSlice is disabled.	
Versioning	Disabled	Transparent cloud tiering has built-in versioning capability, hence IBM Cloud Object Storage versioning can be unavailable. For the Cloud Data Sharing service, versioning might or might not be turned off depending on the needs for retaining versioning on the data.
DeleteRestricted	Yes/No	The gateway does not attempt to delete the vaults, so this setting can be set to yes or no.
Name Index	Disabled	Disabling this setting can result in improved vault performance.
Recovery Listing	Enabled	For performance reasons, the vault that is used for storing data has Name Index disabled and for searchability reasons, the other vault has index that is enabled. On the second provisioning template, Name Index is enabled and the rest of the settings are the same as above.

Essentially, IBM Cloud Object Storage needs two provisioning templates. One of them is used for storing data and the other one is used for metadata or book-keeping. This vault provisioning template must be set as default (Click the **Configure** tab and scroll down to see the option to set the default template). Pass the provisioning code ('demo ' in the example) of the first vault provisioning template to the **mmcloudgateway** command during account creation by using the `--location` parameter.

Firewall recommendations for Cloud services

This topic describes the firewall recommendations that you need to follow to be able to implement Cloud services on your cluster.

Port that can be used is 8085 TCP

- Enables connections to Cloud services nodes from all IBM Spectrum Scale nodes on this port. All communications from non-cluster nodes on this port can be blocked.
- Cloud services nodes are required to communicate with the configured Object storage provider. Typically, this communication occurs over HTTPS (443) or HTTP (80). Contact your Object storage provider for more details.

- The internal port that is used by Cloud services can be changed from 8085 to any other port by using the **mmcloudgateway config** command.

Table 34. Port requirements			
Port number	Protocol	Service name	Component involved in communication
8085	TCP	Transparent cloud tiering	Intra cluster
Object storage provider dependent	TCP	Transparent cloud tiering	Transparent cloud tiering connection to Object storage provider on the external network. Typically HTTPS (443) or HTTP (80)

Note: For firewall recommendations for other components such as performance monitoring tool, protocol access, and GUI, see the *Securing the IBM Spectrum Scale system using firewall* topic in the *IBM Spectrum Scale: Administration Guide*.

Performance considerations

While default configurations on the IBM Spectrum Scale and Cloud services work for the migration and recall of data, some adjustments might be required for optimized performance. This topic addresses some of such adjustments that will likely have the most significant effect on performance.

IBM Spectrum Scale cluster hardware

Memory, CPU, storage, and network play a significant role in determining how well the cloud tiering is going to perform.

Memory: Cloud services requires at least 2 GB RAM for the heap memory, and the IBM Spectrum Scale page pool should also be reasonably high. It is recommended to have a minimum of 64 GB RAM for page pool on the Gateway nodes.

CPU: Cloud services can run more threads to transfer data if the system has more CPU cores available. For better performance, it is recommended to have a minimum of 24 or 32 CPU cores (the more, the better) on the Cloud services gateway nodes.

Storage: The read/write performance from the IBM Spectrum Scale file system is one of the major factors that determine the migrate/recall performance. To achieve desirable performance, the read/write throughput should be more than the available network bandwidth to the cloud. Some of the general recommendations are:

- Have flash disks for storing metadata on the file system.
- If possible, have a separate flash-based file system to keep the Cloud services database. This will ensure that the internal cloud service operations that can span the entire database (like reconcile) run in a reasonable amount of time.

Note: Adding flash storage helps particularly with scaling up the number of files that are supported by Cloud services and is apparent when you get to the hundreds of millions of files migrated to the cloud. So, this recommendation on flash storage only applies if you are scaling to those kinds of numbers.

Network: It is recommended to use 10 GB Ethernet for the network interfaces that are used to transmit data to the cloud. In case of on-premise object store, the object store also should have 10 GB Ethernet. It is recommended to have a dedicated external links to the cloud storage and not to use these interfaces for intra-cluster communication.

Software: Cloud services uses the Data Management Application programming Interface (DMAPI) of the IBM Spectrum Scale to read and write the data from the file system. As Cloud services can migrate data

only as fast as it might read from the file system, it is important to ensure that the read/write performance of the file system is good. Set the following key configuration parameters by using the **mmchconfig** command:

- `pagepool` – 16 GB
- `maxGeneralThreads` – 1280
- `dmapiWorkerThreads` – 64
- `maxFilesToCache` – 100k
- `workerThreads` – 1024

Another important IBM Spectrum Scale configuration is the file system block size. It is recommended to have the block size greater than the Cloud services slice size (default 512 KB). This will ensure that you do not read from the disk more than once to fill in a single slice.

Cloud services configurations

Cloud services has configurable parameters, which can be adjusted based on the environment:

- `connector.server.migrate-threadpool-size` – This determines the number of internal threads that accept the migrate requests from the command line. The default value is 32, and is ideal for most of the situations where you have up to 32 CPU cores. If you have a gateway node with more than 32 CPU cores, it is advisable to increase the number to match the number of CPU cores that you have, so Cloud services can process more migrate requests at once.

Note: This recommendation is for Intel servers. For Power LE servers, you can efficiently run two threads per CPU core, so adjust your thread count accordingly.

- `connector.server.recall-threadpool-size` – Normally you do not adjust this parameter. This determines the number of internal threads that accept recall requests from the command line. If you are sure of your workload, you can adjust the migrate and recall thread pool size. For example, if you have a very cold archive application and expect more migrates and a very limited number of recalls, then you could increase the migrate thread pool size and reduce the recall thread pool size.

For more information, see the *Tuning Cloud services parameters* topic in the *IBM Spectrum Scale: Administration Guide*.

Recommendations on number of Gateway nodes: While increasing the number of gateway nodes to migrate data can increase the total performance, remember that you must find an optimum limit. For example, if the network link from each node to the cloud is 10 Gb and overall read throughput of the file system is 2 GB/s on cluster, just two gateway nodes could achieve the maximum throughput possible. Depending on the file system read/write throughput and the network link, you must make an informed decision on what should be the optimal number of gateway nodes.

Recommendation on number of Cloud Storage Access Points (CSAPs): If you are going to a public cloud service such as the public IBM Cloud Object Storage (COS) service, there is no need to add in multiple access points since these services have a built-in load balancer. For on-premise object storage, you may choose to add in multiple Cloud Storage Access Points (CSAP) if you are not using a load balancer as follows:

- If there are multiple gateway nodes in the cluster, using more than one CSAP for a Cloud Account is a good idea. The number of requests that can be processed by a single Cloud Storage Access Point is limited, and multiple nodes hitting the same access point can cause the performance to drop. A general guide line is to have about as many CSAPs as the number of gateway nodes.

Mixing migrate and recall workloads

Cloud services internally shares threads that carry out an operation on each slice of data, as well as threads that carry out the interactions with cloud object storage. Hence, running a huge workload of migrates and recalls simultaneously, would affect the total turn-around of each activity, compared to running migrates and recalls separately. Cloud services expects migrates to work in batches, where policy identifies a set of cold files to be moved to a cloud storage tier. Recalls, however, are expected to be accessed on demand and are given higher priority than migrates. It is wise to schedule migration policies where possible during off-peak hours to avoid recall performance impacts.

Note: Bulk policy-based recall operations are supported, but not expected to be the common method for recalling files. Since recalls are generally expected to be on demand, recalls are prioritized ahead of migrates.

Distributing files using policies

IBM Spectrum Scale policies can be used to distribute the migrations and recalls to all the nodes in the node class. To effectively use the policy, refer to the following points:

- The **mmapplypolicy** takes two parameters for the number of parallel threads processing the files (-m) and number of files processed per thread (-b).
 - -m has a default value of 24. If you have more CPU cores available, it is advisable to adjust the -m value accordingly
 - -b has a default value of 100. You can increase the number if you want to process more files in one iteration.
 - Plan the -m and -b carefully, so that all the nodes are given equal number of files. For example, if you have 5000 files to migrate and 4 nodes with Cloud services installed, the default values of -m 24 and -b 100 will try to distribute 2400 files to each node. There will not be enough files to be distributed to all nodes, and two nodes will be idle.
- -N parameter of the **mmapplypolicy** will accept the node class name, so the policy will distribute the files to all the nodes in the node class.

Using multiple file system or file sets for the same node group

A customer can configure multiple file systems and multiple file sets for a single node group, and run migrates in parallel for all of them. If they are looking for high performance, it is recommended that they configure the policy in a manner that, migrates run on a single file system or file set at a time. Cloud services design uses a single server process, and resources such as memory, networking and threading are shared between the containers. Running parallel migrates can split the resource, and hence the effective throughput of a single container can be impacted.

Migration policy runs should not be done during the weekly and daily maintenance windows as migrations conflict with some activities that lock the Cloud directory for long periods of time.

Container spillover

For achieving better performance and ease of maintenance, it is recommended to keep no more than 100 million entries per container. After that, the customer must create a new container for the same file system/fileset and continue migrations. This will ensure that a new database is created for the new container, and the container cloud database size does not grow too large. Operation such as reconcile for a 100 million file database can take a few hours. Growing the size much beyond that results in maintenance operations taking too long since the duration of these operations relative to database size is non-linear.

Note:

The automatic container spillover (auto-spillover) is enabled by default during container creation. Auto-spillover is applicable for containers used with tiering service only. Auto-spillover is not applicable for containers used with sharing service. After reaching the default 100 million entries in a container, as an administrator you need not manually create the new container for the same file system any more. When number of entries in an Active container reaches the given threshold value, a new spillover container is automatically created by the Cloud service during the next reconcile operation on container. This container is made an Active container for the configured file system or filesset, for subsequent data migrations. An index value is used as a suffix for new container name. For example, when a container named testContainer reaches spillover threshold value, a new container named testContainer001 is created automatically by the Cloud service. The next spillover index can be seen in the output of **mmcloudgateway containerpairset list**.

Container level auto-spillover settings can be tuned using **mmcloudgateway containerpairset create / update**. More details of auto-spillover CLI options are available under **mmcloudgateway help**. You can create a new container for the given file system at anytime, even before reaching configured

threshold entries per container. Auto-spillover occurs only when the Cloud service finds an active container crossing the configured threshold entries.

Object storage configurations

Use of a load balancer

Object stores such as OpenStack Swift and IBM Cloud Object Storage (formerly known as Cleversafe®) give you an option to have more than one node that can accept the requests from the client. While using a single CSAP, Cloud services can send the requests to only one cloud end-point URL. You can overcome this restriction by using a load balancer to distribute the requests to multiple endpoints.

- NGINX(<https://www.nginx.com/>) - IBM Cloud Object Storage recommends the use of NGINX with COS, and has a solution guide, which can be used to configure NGINX with COS. For most customers, this load balancer does not perform as well as the native load balancer provided by Cloud services.
- HAProxy (<http://www.haproxy.org/>) can be configured on a system, and used to send the requests to more than one end point
- You can use a simple DNS-based load balancer where each DNS lookup for a host name returns an IP address for a different end-point. The host name can be used to create an account with the Cloud services.

If you would prefer that Cloud services distribute the requests to multiple cloud storage endpoint URLs, you can configure more than one CSAP, and Cloud services uses a very basic load balancing algorithm to distribute load across those CSAPs.

IBM Cloud Object Storage

For on-premise configurations, it is recommended to have multiple accessors setup, so that a load balancer or Cloud services itself will have no single point of failure in its cloud services with COS. For public IBM COS service, there is a built-in load balancer – no additional CSAP or load balancer work is necessary. For public IBM COS service, there is a built-in load balancer, so no additional CSAP or load balancer work is necessary.

Swift

For on-premise configurations, it is recommended to have multiple accessors setup, so that a load balancer or Cloud services itself will have no single point of failure in its cloud services with Swift.

Amazon S3

Use the appropriate supported Amazon S3 region while configuring the cloud account. Load balancing or multiple CSAPs are not necessary as Amazon S3 has a built-in load balancer.

Security considerations

You can integrate Cloud services with IBM Security Key Lifecycle Manager (ISKLM) to provide security to the data that is stored on the cloud storage tier, or you can use the native key manager provided with the Cloud services.

Transparent cloud tiering: For information on integration of ISKLM with Transparent cloud tiering, see *Configuring Cloud services with SKLM in IBM Spectrum Scale: Administration Guide*.

Note: Ensure that you back up your security keys by using the **mmcloudgateway service backupConfig** command. Data encrypted by using the Cloud services cannot be recovered, if security keys are lost.

Cloud data sharing: Cloud data sharing currently supports importing any cloud data, assuming it is not read in an encrypted format. Cloud data sharing supports encrypted data only if it was exported by Cloud data sharing and the encryption keys are shared between the importer and the exporter. For this reason, it is recommended that the native Cloud services and Cloud data sharing encryption are used to provide

encryption at rest. A secure connection should be used to transfer data between IBM Spectrum Scale and the cloud. Thus, data is stored encrypted on IBM Spectrum Scale, and stored encrypted on the cloud, and is secured by the connection when transferring between the two systems.

Planning for maintenance activities

Cloud services has five key maintenance activities that need to take place. Some are for keeping normal day-to-day operations going and some are contingencies for service restoration in case of a disaster.

The following two activities are not automated and must be addressed by the administrator as follows:

- Backing up the key library manager - a new copy needs to be made every time there is any change to a key in the key manager. For more information, see the *Backing up the Cloud services configuration* topic in the *IBM Spectrum Scale: Administration Guide*.

Note: If the key manager is lost, there is no backdoor or secret IBM internally known way to recover the data from cloud storage. So, it is important to have a backup copy of the key manager.

- Cloud services leverages SOBAR as a backup mechanism of Transparent cloud tiering metadata that can be used to restore the Transparent cloud tiering service in case of a failure. A sample script is provided in the Transparent cloud tiering directory that can be deployed to run the backups. For more information, see the *Scale out backup and restore (SOBAR) for Cloud services* topic in the *IBM Spectrum Scale: Administration Guide*.

The following three activities are provided for by an automated Cloud services maintenance service:

- Background removal of deleted files from the object storage. This is recommended to be done daily.
- Backing up the Cloud services full database to the cloud. This is recommended to be done weekly.
- Reconciling the Transparent cloud tiering database. This is recommended to be done every four weeks.

The Transparent cloud tiering maintenance service comes with default daily (at night) and weekly (on the weekend) maintenance windows where these activities run. You can use the **mmcloudgateway maintenance status** command to query what your current maintenance window is. It is important to avoid running migration during these activities so you need to make sure your migration policies run outside of those windows. There are scaling considerations when setting up your maintenance windows. They must be long enough so that maintenance activities fit inside the windows. Consider the following guidelines when you plan these activities:

- Reconcile is by far the longest activity and the main one to consider when you plan your service windows. A reconcile for every 100 million file container (which is the default spillover value) takes a few hours if you run metadata with flash. If you run it from a disk, it takes more like 6 to 12 hours.
- Each Transparent cloud tiering node can run one service activity at a time during the maintenance window. For example, if you have three Transparent cloud tiering nodes, you might be running three maintenance activities in parallel at the same time.
- Keep in mind that maintenance activities run to completion even after the maintenance window has passed. The longest duration outside the maintenance window would be an activity that was scheduled and started just prior to the maintenance window closing. That activity would run almost entirely after the maintenance window had completed.

Here is an explanation of how the maintenance window size affects the ability to scale number of Transparent cloud tiering files. If you have a weekly maintenance window of 24 hours, each node would be able to process two-to-three 100 million file containers per week. Since reconcile maintenance needs to be done every 4 weeks, it would follow that a 24-hour maintenance window can accommodate eight-to-twelve 100 million file containers per node every week. With a full 4-node setup, this adds up to 32 - 48 containers (roughly, that is support for 3-5 billion files) per 24-hour maintenance window for a single node group.

It follows that if you want to use Transparent cloud tiering with more than this, you are likely going to want to consider putting the IBM Spectrum Scale metadata and the Transparent cloud tiering database in flash storage. This will greatly increase the number of files the maintenance window can handle.

For setting up maintenance activities, see the *Setting up maintenance tasks* topic in the *IBM Spectrum Scale: Administration Guide*.

Backup considerations for Transparent cloud tiering

You must adhere to some guidelines for backing up files that need to be migrated to a cloud storage tier.

In many cases, Transparent cloud tiering (cooler data) might be run along with file system backups (for hot and warm data) for data protection. Transparent cloud tiering should not be used as a replacement for a valid backup strategy. Files that are migrated with Transparent cloud tiering might be lost if accidentally deleted or if there is a complete site failure. File system backups can protect from these types of events.

When you work with Transparent cloud tiering file system backups, it is important that a file is backed up before it is sent to a cloud storage. A file that is not backed up is recalled from the cloud storage tier when a backup program is run. This can cause unnecessary recall traffic, and also cause the backup to take longer to run. Recalling data also involves cost if you are using a public cloud provider.

To ensure that files are not migrated before they are backed up, it is best to define a migration policy that explicitly excludes files that are recently modified. For example, if nightly backups are run, at minimum, a migration policy should exclude files that are modified within the past day, so they can be backed up before they are migrated. In most cases, it is best to wait longer to avoid recalls if a backup window is missed for some reason. For example, if you run a nightly backup, it is advisable to exclude files that are modified in the past 3 days. If there is an issue with running backups for a long period, it might be necessary to disable a migration policy until the problem with backups is resolved. This avoids many recalls during the next successful backup.

The following is a sample migration policy and it should be tuned as per the backup frequency that is configured for a IBM Spectrum Scale file system.

Note: The amount of time to wait to migrate a file should be greater than the frequency of file system backups.

For information on backing up configuration data, see *Backing up the Cloud services configuration* topic in *IBM Spectrum Scale: Administration Guide*.

For information on backing up the cloud database, see *Backing up the Cloud services database to the cloud* topic in *IBM Spectrum Scale: Administration Guide*.

You can exclude recently modified files with a policy statement such as the following:

```
WHERE (DAYS(CURRENT_TIMESTAMP) - DAYS(MODIFICATION_TIME)) > 5
```

This statement can also be added for manual application of the policy for a directory.

```
/* Define a cloud pool, using the 'mmcloudgateway' command to perform migrations to the pool */
RULE EXTERNAL POOL 'cloudpool' EXEC '/usr/lpp/mmfs/bin/mmcloudgateway files' OPTS '-F'

/* Define a migrate rule, in this example we will migrate from the system pool */
RULE 'MigrateToCloud' MIGRATE FROM POOL 'system'
/* Define the threshold to use, in this example we will migrate when the pool is 85% full,
and attempt to migrate files until the pool is 75% full */
THRESHOLD (85, 75)
/* Next, define a weight for the file to determine which files to migrate first */
/* Here we will use the last access time for the file by subtracting it from the current time,
giving a higher weight to older files */
WEIGHT(CURRENT_TIMESTAMP - ACCESS_TIME)
/* Define that we want to move to the cloud pool */
TO POOL ('cloudpool')
/* We don't want to migrate files that are smaller than 8k in size, because there are little to
no space savings due to file overhead */
WHERE (KB_ALLOCATED > 8)
/* Do not migrate Cloud services internal files, which reside in the .mcstore directory */
AND NOT(PATH_NAME LIKE '%/.mcstore/%')
/* Ignore files that have been modified in the past 5 days. This needs to be customized based on
```

```
the backup frequency. */
AND (DAYS(CURRENT_TIMESTAMP) - DAYS(MODIFICATION_TIME)) > 5
```

Quota support for tiering

Transparent cloud tiering supports container-level quotas in ways that are described here.

IBM Spectrum Scale does not account for data that Transparent cloud tiering has migrated to the Cloud for quota support.

- Files in non-resident states do not count against quota as defined by IBM Spectrum Scale.

Object storage generally supports quotas and might be used to provide quota support.

- For example, IBM Cloud Object Storage supports hard quotas at the granularity of a container.

General guidelines for quota support:

- Use fileset support – this provides the level of granularity you need for cloud storage quotas to be useful.
- Set the desired quota for each fileset by using the object storage quota setting for the container.
- On container spillover, the size for each container in the fileset needs to be accounted for.
 - Since all but the most recent container will no longer be growing, you can set the quota for the most recent (and active) container based on the remaining capacity after subtracting for the current size of the older containers.
 - If older containers have a significant number of files deleted from them, the quota for the active container might need to be periodically adjusted upwards.
- A quota support script is available in the samples directory. that provides a report of space taken in the cloud for a given filesystem. The report does not include space allocated for files that are deleted or are overwritten with a new version that are being retained for the specified retention period.

Note: For more information on setting quota at a container level, see documentation for the specific cloud account that you use in your environment.

Client-assisted recalls

Non-gateway nodes can have an extended client which can service the recalls locally.

You must install a transparent cloud tiering server package on a non-gateway node.

To enable recalls from a non-gateway node, run **mmcloudgateway clientassist enable** on the non-gateway node where the package is installed. If a node is client-assist enabled, the transparent cloud tiering server nodes might still be called upon for transparent recalls if the node gets overloaded with transparent recall requests with a full queue.

All recalls originating from this node are serviced from this node only, without routing through the gateway node.

To disable client assisted recalls, run **mmcloudgateway clientassist disable**.

Planning for AFM

The following topics assist you in planning for AFM.

Requirements for UID and GID on the cache and home clusters

This topic describes the UID and GID requirements on the cache and home clusters.

User IDs and group IDs must be managed the same way across the cache and home clusters. However, for AFM relationships, which are using the native GPFS protocol, where user IDs are different on the home and the cache, you can remap the IDs by using the GPFS UID remapping. For more information about ID mapping, see *Configuring ID mappings in IMU* in *IBM Spectrum Scale: Administration Guide*.

Recommended workerThreads on a cache cluster

Specifies the number of **workerThreads** on a cache cluster.

It is recommended to increase the **workerThreads** to twice the default value while you use GPFS backend. For NFS protocol, the number of **workerThreads** can be equal to the default value. The default value is 48.

Table 35. workerThreads value per backend	
Backend	Minimum workerThreads value
NFS	48 (default)
GPFS	96 (2*default)

Inode limits to set at cache and home

To control the number of inodes that are used by a cache or home fileset, you can specify a limit on the number of inodes for that fileset during fileset creation time. For more information about the `--inode-limit` option, see *mmcrfileset command* in *IBM Spectrum Scale: Command and Programming Reference*.

If a home is a dependent fileset in the file system, the inode limit used are the ones for the independent fileset where the dependent fileset is located. If the dependent fileset is located in the "root" fileset then the file system limits are used. However, if the dependent fileset is located in a independent fileset that is not the "root" fileset then the inode limits for that independent fileset are used.

Note: AFM does not perform eviction of file inodes, even if the inode limit is reached. The cache eviction is applicable only for file data block quotas.

Planning for AFM gateway nodes

An AFM cache cluster must include minimum one node running the Linux operating system. This node can be assigned the gateway nodes role for the AFM cache filesets. Every AFM fileset is assigned to a unique AFM gateway, which is called as a primary gateway. This gateway handles all the communication between the AFM fileset and the target on the home site for synchronization.

You can assign and remove a gateway node role by using the following commands:

- To assign gateway node role to the cluster member node, issue the following command:

```
mmchnode -N node1,node2,... --gateway
```

- To remove the gateway node role from the cluster member node, issue the following command:

```
mmchnode -N node1,node2,... --nogateway
```

The AFM gateway node is licensed as a server node.

Multiple gateway nodes

An IBM Spectrum Scale cluster can have one or more gateway nodes. Multiple gateway nodes are helpful if you want to assign different filesets to a different gateway node and configure parallel data transfer. The assignment of the gateway node to the AFM cache fileset is based on the Hashing Algorithm. You can also assign a gateway node to an AFM cache fileset.

For more information about Parallel Data Transfer, see [“Parallel data transfers” on page 59](#).

AFM is enabled with IBM Spectrum Scale Data Access Edition.

For more information about the licensing, see [“IBM Spectrum Scale product editions” on page 213](#).

General recommendations for AFM gateway node configuration

To configure an AFM gateway node, the general recommendations are as follows:

- The memory and CPU requirement for a gateway node depends on the number of assigned AFM, AFM-DR fileset, or AFM to cloud object storage fileset and files in the filesets.
- Minimum 128 GB of memory is needed for the gateway node.
- An AFM gateway node can be assigned up to 20 AFM, AFM-DR, or AFM to cloud object storage filesets.
- A gateway node is a dedicated node in the cluster without any other designations such as NSD server, CES protocol nodes, quorum, and manager.
- The recommended number of inodes is approximately 400 million in all AFM, AFM-DR, and AFM to cloud object storage fileset that is served by a single gateway node.

The following table contains the cluster level parameters:

Parameter	Value
Pagepool	8G
afmHardMemThreshold	40G
afmNumFlushThreads	8
afmDIO	2
maxFilestoCache	10000

To set these parameters, use the **mmchconfig** command. For example,

```
# mmchconfig pagepool=8G
```

For more information about configuration parameters, see *Parameters for performance tuning and optimization* in the *IBM Spectrum Scale: Administration Guide*.

All these recommendations are based on observations in a controlled test environment by running moderate or reasonable workload. The parameters values might vary based on the setup or workload.

Planning for AFM DR

The following topics assist you in planning for AFM DR.

Requirements for UID/GID on primary and secondary clusters

User IDs and group IDs must be managed the same way across the primary and secondary clusters.

However, for AFM DR relationships, which use native GPFS protocol, where user IDs are different on the primary and secondary clusters, the IDs may be mapped again by using GPFS UID remapping.

Recommended worker1Threads on primary cluster

IBM recommends that you increase the **worker1Threads** to twice the number of AFM DR filesets that are using GPFS backend, which exist on all file systems in the cache cluster.

For example, if a cluster has 50 primary fileset that is using NFS protocol and 60 primary using GPFS protocol, the **worker1Threads** must be at least 160 ($50 + 2 \times 60$). For NFS protocol, the number of **worker1Threads** can be equal to the number of primary filesets.

NFS setup on the secondary cluster

NFS setup on the secondary cluster follows the same rules as the NFS setup on the AFM home cluster.

General guidelines and recommendations for AFM-DR

AFM-DR is a disaster recovery solution, which is by nature a business-critical solution. Its success depends on administration discipline, including careful design, configuration, and testing. Considering this, IBM requires that the AFM Async DR (AFM-ADR) deployment must follow certain guidelines and recommendations for the deployment. The AFM-ADR feature is disabled by default and it can be enabled by following the recommendations and guidelines.

However, the general recommendations and prerequisites for the AFM-DR setup for general usage might differ for use-case specific configurations. The general recommendations help to understand the requirement better for the AFM-DR deployment.

Guidelines or recommendations for the deployment of IBM Spectrum Scale AFM-DR are as follows:

AFM-DR filesets and files recommendation

- The number of AFM DR filesets that can be configured in an IBM Spectrum Scale cluster is up to 100.
- The number of files in an IBM Spectrum Scale AFM-ADR environment is approximately 100 million files in every fileset.

Gateway nodes recommendation

- The number of gateway nodes that can be configured in an IBM Spectrum Scale cluster depends upon the number of AFM filesets.
- Keep AFM gateway nodes equal to 1/10th of the number of filesets. That is, for 20 ADR filesets in a primary cluster, the recommended number of gateway nodes is '2'. So that, every gateway can be assigned up to 10 filesets.
- The gateway node must have a dedicated role and no other role must be assigned to the gateway node.
- Gateway node(s) are needed at primary Site only. AFM or AFM-DR do not need any Gateway node at the secondary site.

Memory and storage recommendation for a gateway node

- Random access memory (RAM) on the gateway node should be 128 G.
- The `/var` partition space storage needs to be provisioned for bigger storage (based on the number of filesets handled by gateway node * files in fileset * 255 bytes) space for internal usage during the recovery or resynchronization.

Configuration parameters recommendation for AFM-ADR

Tunable	Recommended value
Pagepool	8G
afmHardMemThreshold	40G
afmNumFlushThreads	8
afmDIO	2
maxFilesToCache	10000
afmMaxParallelRecoveries	3

To set these parameters, use the **mmchconfig** command. For example,

```
# mmchconfig pagepool=4G
```

For more information about configuration parameters, see *Parameters for performance tuning and optimization* in the *IBM Spectrum Scale: Administration Guide*.

Network and computing requirement

- Plan necessary network bandwidth between primary and secondary clusters to replicate incoming changes based on a workload.
- This requirement varies based on the workload. The application is slower on AFM DR filesets because of the message queuing on the gateway node for the replication. This performance hit varies from 1.2x to 4x based on network and gateway node resources. If the gateway node handles many filesets, more computational power is required for filtering and dependency checking logic.

Data synchronization for the AFM fileset conversion

- You can convert an existing GPFS-independent fileset to an AFM-ADR fileset by using the `--inband` trucking method. You must not use the `--outband` (deprecated) option even though data was copied by using the external tool.
- If the secondary site has already data, AFM does not copy data during the initial resynchronization. You can still truck (out of band) data by using any other tool such as `rsync` and convert the fileset by using the `--inband` option. AFM expects that files are in sync with nanoseconds time difference. Use `rsync` version `>= 3.10` on both source and destination to copy the data and later use the `inband` conversion. The `inband` conversion skips the data, if data exists (and matches) on the secondary site.

Limitations

For more information about the limitations of AFM-DR, see [“AFM and AFM DR limitations” on page 95](#).

Planning for AFM to cloud object storage

The considerations for the planning AFM to cloud object storage relation are as follows:

The AFM to cloud object storage relation is between an AFM to cloud object storage fileset that is hosted on an IBM Spectrum Scale cluster and cloud object storage bucket, or a directory within an AFM to cloud object storage fileset and a cloud bucket. For more information, see [“AFM to cloud object storage operation modes” on page 119](#). The important elements that need consideration in the IBM Spectrum Scale cluster are the host system, storage, operating system, and networking. For more information, see [“Planning for GPFS” on page 219](#).

Before you set the AFM to cloud object storage relation, consider the following points:

1. Cloud object storage such as Amazon S3 and IBM Cloud Object Storage must be in accordance with the workload requirement. Access keys and secret keys that are provided by these services need to be set up for AFM to cloud object storage filesets.

To transfer the data between AFM to cloud object storage and cloud object storage endpoints, select an appropriate protocol, for example, HTTP or HTTPS.

2. On an IBM Spectrum Scale cluster, file system space and inodes (number of inodes) need to be planned for an AFM to cloud object storage fileset first before provision. For more information, see [“File system creation considerations” on page 251](#).

To control the number of inodes that are used by the AFM to cloud object storage fileset, you can specify a limit on the number of inodes for that fileset. To set inodes limit, use the `mmchfileset` command after the `mmafmcconfig` command is run to set the relationship. For more information, see `mmchfileset` command in the *IBM Spectrum Scale: Command and Programming Reference*.

3. Planning of gateway nodes

An IBM Spectrum Scale cluster can have one or more gateway nodes. Multiple gateway nodes are helpful, if you want to assign different filesets to a different gateway node. The multiple nodes on a gateway node are useful to load balance the workload and also helpful for recovery and upgrade scenarios.

Gateway nodes need to have access to cloud object storage endpoints. For more information about gateway node recommendation, see [“General recommendations for AFM gateway node configuration” on page 343](#).

4. Network considerations

Plan necessary network bandwidth between AFM to cloud object storage and IBM Spectrum Scale clusters to replicate data and workload.

The networking element is the major contributor in the overall performance delivery of the IBM Spectrum Scale cluster where a few nodes act as Network Shared Disk (NSD) servers while most of the client nodes access the file system over a TCP/IP network. Gateway nodes in this host system connect to the outside world, which in this case is Amazon S3 cloud object storage or IBM Cloud Object Storage.

All the firewalls are set up to allow the gateway nodes to discover and communicate with these cloud storage endpoints.

5. When you work with temporary credentials for accessing Cloud Object Storage like, Amazon S3, IBM Cloud and other services, you must obtain the security token and create fileset by specifying **--user-keys** option of **mmafmcconfig** command.

The `mmuid2keys (/var/mmfs/etc/mmuid2keys)` file must be configured to provide credentials in the following format as shown:

```
Access_Key:Secret_Key:STS
```

Firewall recommendations for AFM to cloud object storage

Consider these firewall recommendations for the AFM to cloud object storage setup.

To enable communication between a cloud object storage server and IBM Spectrum Scale AFM to cloud object storage, firewall must keep open a specific port or ports on all the gateway nodes. AFM to cloud object storage communicates to a cloud object storage server on a specified port such as 80 and 443. The specified port is provided as a part of an endpoint for a fileset. Different filesets can have a different port. For this communication, a specific port needs to be opened or enabled for the AFM to cloud object storage.

Firewall recommendations

The IBM Spectrum Scale system administrator is supposed to follow certain recommendations to set up firewall to secure the IBM Spectrum Scale system from unauthorized access.

For more information about firewall recommendations, see *Securing the IBM Spectrum Scale system using firewall* in *IBM Spectrum Scale: Administration Guide*.

Considerations for GPFS applications

Application design should take into consideration the exceptions to Open Group technical standards with regard to the `stat()` system call and NFS V4 ACLs. Also, a technique to determine whether a file system is controlled by GPFS is provided.

For more information, see the following topics in *IBM Spectrum Scale: Administration Guide*:

- *Exceptions to Open Group technical standards*
- *Determining if a file system is controlled by GPFS*
- *GPFS exceptions and limitation to NFS V4 ACLs*
- *Considerations for GPFS applications*

Security-Enhanced Linux support

IBM Spectrum Scale runs on Red Hat Enterprise Linux operating systems with Security-Enhanced Linux (SELinux).

IBM Spectrum Scale supports SELinux that is set to one of the available SELinux modes such as enforcing or permissive and the SELinux policy set to a targeted policy. For more information about SELinux support for the Object Storage, see [“SELinux considerations” on page 323](#) for Object.

Note: IBM Spectrum Scale daemons will run in unconfined domains.

Space requirements for call home data upload

Certain space requirements must be met by the system before uploading the call home data. The following section details the space requirements for call home.

The call home component uses the directory, specified in the IBM Spectrum Scale settings variable *dataStructureDump*, for saving the temporary data. By default, this directory is */tmp/mmfs*, but it can be changed by the customer by using the **mmchconfig** command. The current value can be read by executing the following command:

```
mmdiag --config | grep "dataStructureDump"
```

The space requirements differ depending on the use case. The following two situations are possible:

- For uploading daily and weekly packages, and non-PMR or Salesforce case-related files you require space equivalent to two times the size of the file that is to be uploaded. For example, if you need to upload 1 GB of data, then there needs to be at least 2 GB of disk space for the file to be properly uploaded.
- For uploading the ticket-related files, a dynamical buffered chunking system is used. In this case, the minimum space required for an upload is 1/250 of the file size or 0.4% of the size of the file to be uploaded. At least 200 MB of space is recommended to minimize the chunking overhead and maximize the transfer speed.

Chapter 3. Steps for establishing and starting your IBM Spectrum Scale cluster

There are several steps you must perform to establish and start your IBM Spectrum Scale cluster. This topic provides the information you need for performing those steps.

The installation toolkit, available on a few Linux distributions, automates many of the following steps. For more information, see [“Installing IBM Spectrum Scale on Linux nodes with the installation toolkit” on page 386](#).

You can install IBM Spectrum Scale and deploy protocols either manually or by using the installation toolkit. This topic provides the information you need for establishing and starting your IBM Spectrum Scale cluster manually. If you have already installed IBM Spectrum Scale with the installation toolkit, these steps have already been completed.

Tip: Ensure that there is approximately 5 GB of free space on the nodes to download, extract, and install the packages.

Follow these steps to establish your IBM Spectrum Scale cluster:

1. Review supported hardware, software, and limits in the [IBM Spectrum Scale FAQ in IBM Documentation](#) for the latest recommendations on establishing a IBM Spectrum Scale cluster.
2. Install the IBM Spectrum Scale licensed program on your system:
 - For existing systems, see [Chapter 16, “Upgrading,” on page 505](#).
 - For new systems:
 - For your Linux nodes, see [Chapter 4, “Installing IBM Spectrum Scale on Linux nodes and deploying protocols,” on page 351](#).
 - For your AIX nodes, see [Chapter 5, “Installing IBM Spectrum Scale on AIX nodes,” on page 453](#).
 - For your Windows nodes, see [Chapter 6, “Installing IBM Spectrum Scale on Windows nodes,” on page 457](#).
3. Decide which nodes in your system will be quorum nodes (see [“Quorum” on page 223](#)).
4. Create your GPFS cluster by issuing the `mmcrcluster` command. See [“GPFS cluster creation considerations” on page 228](#).
5. Use the `mmchlicense` command to assign an appropriate GPFS license to each of the nodes in the cluster. See [“IBM Spectrum Scale license designation” on page 215](#) for more information.

If you use the installation toolkit to install IBM Spectrum Scale then steps 2 to 5 in the following procedure are completed by the installation toolkit, and step 6 can be optionally completed by the installation toolkit.

After your IBM Spectrum Scale cluster is established:

1. Ensure you have configured and tuned your system according to the values suggested in the *Configuring and tuning your system for GPFS* topic in *IBM Spectrum Scale: Administration Guide*.
2. Start IBM Spectrum Scale by issuing the `mmstartup` command. For more information, see **`mmstartup` command** in *IBM Spectrum Scale: Command and Programming Reference*.
3. Create new disks for use in your file systems by issuing the `mmcrnsd` command. See [“Network Shared Disk \(NSD\) creation considerations” on page 239](#).
4. Create new file systems by issuing the `mmcrfs` command. See [“File system creation considerations” on page 251](#).
5. Mount your file systems by issuing the `mmmout` command.
6. As an optional step, you can also create a temporary directory (`/tmp/mmfs`) to collect problem determination data. The `/tmp/mmfs` directory can be a symbolic link to another location if more space

can be found there. If you decide to do so, the temporary directory should *not* be placed in an IBM Spectrum Scale file system, as it might not be available if IBM Spectrum Scale fails.

If a problem occurs, IBM Spectrum Scale might write 200 MB or more of problem determination data into `/tmp/mmfs`. These files must be manually removed when any problem determination is complete. This should be done promptly so that a **NOSPACE** condition is not encountered if another failure occurs. An alternate path can be specified by issuing the `mmchconfig dataStructureDump` command.

Chapter 4. Installing IBM Spectrum Scale on Linux nodes and deploying protocols

Use this information for installing IBM Spectrum Scale on Linux nodes and deploying protocols.

Before installing IBM Spectrum Scale, you must review [“Planning for GPFS” on page 219](#) and the [IBM Spectrum Scale FAQ in IBM Documentation](#).

Installing IBM Spectrum Scale without ensuring that the prerequisites listed in [“Hardware requirements” on page 219](#), [“Software requirements” on page 220](#), and [“Installation prerequisites” on page 352](#) are satisfied can lead to undesired results.

Note:

- CES supports HDFS protocols. CES HDFS follows the same installation methods and prerequisites as the other protocols as documented in this section. For specific information about installing CES HDFS, see *Installation* under [CES HDFS in Big data and analytics support documentation](#).
- For information about installing IBM Spectrum Scale Erasure Code Edition, see *IBM Spectrum Scale Erasure Code Edition documentation*.

The following methods are available for installing IBM Spectrum Scale on Linux nodes.

Installation method	Description
Installation toolkit	<p>On supported Linux distributions, you can install IBM Spectrum Scale and deploy protocols by using the installation toolkit.</p> <p>For more information, see “Installing IBM Spectrum Scale on Linux nodes with the installation toolkit” on page 386.</p> <p>For information on limitations if you are using the installation toolkit, see “Limitations of the installation toolkit” on page 394.</p>
Manual	<p>You can install the IBM Spectrum Scale packages manually by using commands such as rpm or yum on Red Hat Enterprise Linux, rpm or zypper on SLES, and dpkg or apt on Ubuntu Linux.</p> <p>For more information, see “Manually installing the IBM Spectrum Scale software packages on Linux nodes” on page 359.</p>

Deciding whether to install IBM Spectrum Scale and deploy protocols manually or with the installation toolkit

On supported Linux distributions, you can install IBM Spectrum Scale and deploy protocols either manually or using the installation toolkit.

Why would I want to use the installation toolkit?

While planning your installation, consider the advantages provided by the installation toolkit. The installation toolkit:

1. Simplifies IBM Spectrum Scale cluster creation.
2. Automatically creates NSD and file system stanza files.
3. Creates new NSDs, file systems, and adds new nodes.
4. Has a single package manager for all components of a release.
5. Deploys Object, SMB, NFS, and HDFS by automatically pulling in all prerequisites.

6. Configures file and object authentication during deployment.
7. Consistently sets and synchronizes time on all nodes.
8. Deploys the IBM Spectrum Scale GUI.
9. Configures Performance Monitoring consistently across all nodes.
10. Installs and configures file audit logging and call home functions.
11. Simplifies upgrade with a single command to upgrade all components on all nodes.

Before deciding which method you want to use, review the following topics:

- [“Installation prerequisites” on page 352](#)
- [“IBM Spectrum Scale packaging overview” on page 355](#)
- [“Understanding the installation toolkit options” on page 392](#)
- [“Limitations of the installation toolkit” on page 394](#)
- [“Using the installation toolkit to perform installation tasks: Explanations and examples” on page 407](#)
- [“Manually installing the IBM Spectrum Scale software packages on Linux nodes” on page 359](#)

Installation prerequisites

These are the several prerequisites for installing IBM Spectrum Scale including those for installing protocols and performance monitoring.

Required packages for supported Linux distributions

For the list of packages that must be installed on the system before you can install IBM Spectrum Scale, see [“Software requirements” on page 220](#).

For a list of supported operating Systems, architectures, and kernels, see [IBM Spectrum Scale FAQ in IBM Documentation](#).

Cleanup required if you previously used the Object Redpaper configuration process

If you have already configured the system with OpenStack software to use GPFS (following instructions from the [Object Red paper](#)) be sure to follow the Object cleanup steps in [“Cleanup procedures required if reinstalling with the installation toolkit” on page 496](#) before you start any installation activity.

Same version required on protocol and quorum nodes

If you want to run protocols in a mixed-version cluster, the protocol nodes and all of the quorum nodes must be running on the same version.

Note: For the object protocol, the GPFS software is not required on the node that hosts the Keystone identity service if it is deployed on a separate node from the GPFS protocol nodes.

Additional GPFS requirements

- If you want to run protocols, Cluster Configuration Repository (CCR) must be available.
- `mmchconfig release=LATEST` must be run.
- A GPFS cluster and a file system are required. If this infrastructure does not already exist, you must install the GPFS software, create a GPFS cluster and create a file system. You can use the installation toolkit to do these tasks. For more information, see [“Using the installation toolkit to perform installation tasks: Explanations and examples” on page 407](#).
- The GPFS file system must be mounted on all GPFS protocol nodes.
- It is strongly recommended to configure the file system to only support NFSv4 ACLs. You can use the installation toolkit to do this task also if you use the installation toolkit to install GPFS. For more information, see [“Using the installation toolkit to perform installation tasks: Explanations and examples” on page 407](#).

Alternatively, you can use the `-k nfs4` parameter for `mmcrfs`. NFSv4 ACLs are a requirement for ACL usage with the SMB and NFS protocols. For more information, see `mmcrfs` examples in the *IBM Spectrum Scale: Command and Programming Reference*.

- Quotas are not enabled by default in GPFS file systems but are recommended for use with SMB and NFS protocols. For more information about quota management, see *Enabling and disabling GPFS quota management* in *IBM Spectrum Scale: Administration Guide*.

Creation of a file system or fileset or path for a CES shared root, and creation of an object fileset

The installation toolkit uses a shared root storage area to install the protocols on each node. This storage is also used by NFS and object protocols to maintain system data associated with the cluster integration. This storage can be a subdirectory in an existing file system or it can be a file system on its own. For production systems, it is recommended to create a dedicated file system for this purpose due to performance reasons. Once this option is set, changing it requires a restart of GPFS.

You can use the installation toolkit to set up this CES shared root storage area if you use the toolkit for GPFS installation and file system creation. For more information, see [“Using the installation toolkit to perform installation tasks: Explanations and examples”](#) on page 407.

However, if you want to set up shared root before launching the installation toolkit, the following steps can be used:

1. Create a file system or fileset for the shared root. Size must be at least 4 GB.
2. Use the following command:

```
mmchconfig cesSharedRoot=path_to_the_filesystem/fileset_created_in_step_1
```

For object, the installation toolkit creates an independent fileset in the GPFS file system that you name.

SSH and network setup

The node used for initiating installation of SMB, NFS, and/or Object protocols by using the installation toolkit must be able to communicate through an internal or external network with all protocol nodes to be installed. All nodes also require SSH keys to be set up so that the installation toolkit can run remote commands without any prompts. Examples of prompts include a prompt for a remote node's password or a prompt for a yes-or-no question. No prompts should occur when using SSH among any cluster nodes to and from each other, and to and from the node designated for installation.

For information on ports required by IBM Spectrum Scale, see *Securing the IBM Spectrum Scale system using firewall and GPFS port usage* in *IBM Spectrum Scale: Administration Guide*.

For examples of how to open firewall ports on different operating systems, see *Examples of how to open firewall ports* in *IBM Spectrum Scale: Administration Guide*.

Repository setup

The installation toolkit contains all necessary code for installation. However, for manual installation or installation with the installation toolkit, there might be base operating system packages required as prerequisites. To satisfy any prerequisites, it is necessary for your nodes to have access to a DVD repository or an external repository accessible by network. Repositories containing IBM Spectrum Scale dependencies include the following `x86_64` example:

```
rhel-x86_64-server-7 Red Hat Enterprise Linux Server
```

For information on setting up repositories on Red Hat Enterprise Linux nodes, see [Configuring Yum and Yum Repositories](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/sec-Configuring_Yum_and_Yum_Repositories.html) (https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/sec-Configuring_Yum_and_Yum_Repositories.html).

For information on setting up repositories on SLES nodes, see [Managing Software Repositories and Services](https://documentation.suse.com/sles/15-SP1/html/SLES-all/cha-yast-software.html#sec-yast-software-instsource) (<https://documentation.suse.com/sles/15-SP1/html/SLES-all/cha-yast-software.html#sec-yast-software-instsource>).

For information on setting up repositories on Ubuntu nodes, see [Repositories/Ubuntu](https://help.ubuntu.com/community/Repositories/Ubuntu) (<https://help.ubuntu.com/community/Repositories/Ubuntu>).

Important: You must disable all configured EPEL repositories on all nodes added to the installation toolkit before proceeding with installation, deployment, or upgrade.

Network time protocol (NTP) setup

You must configure time synchronization on all nodes in your cluster by using tools such as NTPD and Chrony. IBM Spectrum Scale necessitates all clocks to be in sync for installation, deployment, upgrade, and cluster operation. For information about using these tools with the operating system on your nodes, see the respective operating system documentation.

Collection of core dump data

For information about changing configuration to enable collection of core dump data from protocol nodes, see *Configuration changes required on protocol nodes to collect core dump data* in *IBM Spectrum Scale: Problem Determination Guide*.

Preparing the environment on Linux nodes

Before proceeding with installation, prepare your environment by following the suggestions in the following sections.

Add the GPFS bin directory to your shell PATH

Ensure that the PATH environment variable for the root user on each node includes `/usr/lpp/mmfs/bin`. (This is not required for the operation of IBM Spectrum Scale, but it can simplify administration.)

Other suggestions for cluster administration

GPFS commands operate on all nodes required to perform tasks. When you are administering a cluster, it can be useful to have a more general form of running commands on all nodes. A suggested way to do this is to use an OS utility like `dsh` or `pdsh` that can execute commands on all nodes in the cluster. For example, you can use `dsh` to check the kernel version of each node in your cluster:

```
# uname -opr
3.10.0-693.el7.x86_64 x86_64 GNU/Linux
```

Once you have `dsh` set up, you can use it to install IBM Spectrum Scale on a large cluster. For details about setting up `dsh` or a similar utility, review the documentation for the utility.

Verify that prerequisite software is installed

Before installing IBM Spectrum Scale, it is necessary to verify that you have the correct levels of the prerequisite software installed on each node in the cluster. If the correct level of prerequisite software is *not* installed, see the appropriate installation manual before proceeding with your IBM Spectrum Scale installation.

Installing IBM Spectrum Scale without ensuring that the prerequisites listed in [“Hardware requirements” on page 219](#), [“Software requirements” on page 220](#), and [“Installation prerequisites” on page 352](#) are satisfied can lead to undesired results.

For the most up-to-date list of prerequisite software, see the [IBM Spectrum Scale FAQ in IBM Documentation](#).

The FAQ contains the latest information about the following:

- Supported Linux distributions and kernel levels
- Recommended or required RPM levels
- Software recommendations
- Configuration information

Before proceeding, see also [“GPFS and network communication” on page 15](#).

IBM Spectrum Scale packaging overview

The IBM Spectrum Scale self-extracting package consists of several components.

Components	Description
Installation toolkit and license	The components necessary to begin the protocol installation.
GPFS	Core GPFS is included in this self-extracting package. It also includes call home and file audit logging packages.
NFS (Ganesha)	The packages for Ganesha, the open-source, user-space implementation of the NFS protocol. They can be installed, enabled, and configured using the installation toolkit.
SMB (Samba)	The packages for Samba, the open-source implementation of the SMB protocol. They can be installed and enabled using the installation toolkit.
Object (Swift and Keystone)	The Swift and Keystone packages necessary for enablement of Object services. They can be installed, enabled, and configured using the installation toolkit.
Performance monitoring tool	The packages necessary for performance monitoring by GPFS (including protocols). They can be installed and enabled using the installation toolkit.

Preparing to install the IBM Spectrum Scale software on Linux nodes

Follow the steps in this topic in the specified order to prepare to install the IBM Spectrum Scale software.

Accepting the electronic license agreement on Linux nodes

The GPFS software license agreement is shipped with the GPFS software and is viewable electronically. When you extract the GPFS software, you are asked whether or not you accept the license.

The electronic license agreement must be accepted before software installation can continue. Read the software agreement carefully before accepting the license. See [“Extracting the IBM Spectrum Scale software on Linux nodes”](#) on page 355.

Extracting the IBM Spectrum Scale software on Linux nodes

The IBM Spectrum Scale software is delivered in a self-extracting archive. This self-extracting archive can be downloaded from [Fix Central](#).

The self-extracting image contains the following.

- The IBM Spectrum Scale product installation SLES and Red Hat Enterprise Linux RPMs and Ubuntu Linux packages
- The License Acceptance Process (LAP) tool

The LAP tool is invoked for acceptance of the IBM Spectrum Scale license agreements. The license agreements must be accepted to obtain access to the IBM Spectrum Scale product installation images.

- A version of the Java Runtime Environment (JRE) necessary to run the LAP tool

1. Copy the self-extracting product image to a local directory, specifying the correct version of the product for your hardware platform and Linux distribution. For example:

```
cp /media/archive/Spectrum_Scale_Standard-5.1.5.x-x86_64-Linux-install/  
/tmp/Linux/Spectrum_Scale_Standard-5.1.5.x-x86_64-Linux-install
```

2. Verify that the self-extracting program has executable permissions, for example:

```
# ls -l /tmp/Spectrum_Scale_Standard-5.1.5.x-x86_64-Linux-install
```

The system displays information similar to the following:

```
-rwxr-xr-x 1 root root 673341648 Nov 10 13:53 /tmp/Spectrum_Scale_Standard-5.1.5.x-x86_64-Linux-install
```

If it is not executable, you can make it executable by issuing the `chmod +x` command.

Note: To get a list of all packages in the self-extracting installation image, issue the following command (specifying the `--manifest` option):

```
Spectrum_Scale_Standard-5.1.5.x-x86_64-Linux-install --manifest
```

A sample output is as follows:

```
manifest
sw, rpm, gpfs.base-5.1.5-x.x86_64.rpm,
Thu 16 Jun 2020 10:40:41 AM MST, md5sum: 99e940728f73649520533c06ae14b941
sw, rpm, gpfs.adv-5.1.5-x.x86_64.rpm,
Thu 16 Jun 2020 10:46:43 AM MST, md5sum: 8c7a635ee2381b075e7960813bef3c47
sw, rpm, gpfs.gskit-8.0.55.x.x86_64.rpm,
Thu 16 Jun 2020 10:45:15 AM MST, md5sum: 2e844aa5351dd87c234f5273ef9f3673
sw, rpm, gpfs.gss.pmsensors-5.1.5-x.el7.x86_64.rpm,
Wed 15 Jun 2020 09:56:29 AM MST, md5sum: 3f4c084e93f3bd0d7cdf4adb8fe65448
sw, rpm, gpfs.gss.pmc collector-5.1.5-x.el7.x86_64.rpm,
Wed 15 Jun 2020 09:56:30 AM MST, md5sum: 5fd3d21c935fea27d3b8c9b97b8c49e2
sw, rpm, gpfs.gss.pmsensors-5.1.5-x.sles12.x86_64.rpm,
Wed 15 Jun 2020 09:56:34 AM MST, md5sum: abab4976a83814877a140bab5e96c12f
sw, rpm, gpfs.gss.pmc collector-5.1.5-x.sles12.x86_64.rpm,
Wed 15 Jun 2020 09:56:31 AM MST, md5sum: 48ae6aedf41c6681c98bc167dc1babdf
sw, rpm, gpfs.gpl-5.1.5-x.noarch.rpm,
Thu 16 Jun 2020 10:52:11 AM MST, md5sum: 5989f71ace5dd4e7ca92f030de451819
sw, rpm, gpfs.gui-5.1.5-x.noarch.rpm,
Mon 13 Jun 2020 10:34:37 AM MST, md5sum: e562910893e468cf9790865093623129
sw, rpm, gpfs.msg.en_US-5.1.5-x.noarch.rpm,
Thu 16 Jun 2020 10:44:37 AM MST, md5sum: 6fe50b1bf9265d6dab322e0a472266ec
```

(The remaining packages in the installation image are displayed.)

3. Invoke the self-extracting image that you copied to a local directory by using a command such as `Spectrum-Scale-Standard-5.1.5.x-x86_64-Linux-install` and accept the license agreement:

- a. By default, the LAP tool, JRE, and IBM Spectrum Scale installation images are extracted to the target directory `/usr/lpp/mmfs/5.1.5.x`
- b. The license agreement files on the media can be viewed in text-only mode or graphics mode:
 - Text-only is the default mode. When run the output explains how to accept the agreement:

```
<...Last few lines of output...>
Press Enter to continue viewing the license agreement, or
enter "1" to accept the agreement, "2" to decline it, "3"
to print it, "4" to read non-IBM terms, or "99" to go back
to the previous screen.
```

- To view the files in graphics mode, invoke `Spectrum-Scale-Standard-5.1.5.x-x86_64-Linux-install`

Using the graphics-mode installation requires a window manager to be configured.

- c. You can use the `--silent` option to accept the license agreement automatically.
- d. Use the `--help` option to obtain usage information from the self-extracting archive.

Upon license agreement acceptance, the IBM Spectrum Scale product installation images are placed in the extraction target directory (`/usr/lpp/mmfs/5.1.5.x`). This directory contains the IBM Spectrum Scale packages according to the target Linux distribution that are applicable for the IBM Spectrum Scale edition that you installed (Standard, Advanced, or Data Management). For more information, see [“Location of extracted packages” on page 361](#).

Note: For this release, the IBM Global Security Kit (GSKit) version for RPMs and Ubuntu Linux packages must be at least 8.0.55.x or higher.

In this directory there is a `license` subdirectory that contains license agreements in multiple languages. To view which languages are provided, issue the following command:

```
ls /usr/lpp/mmfs/5.1.5.x/license
```

The system displays information similar to the following:

```
Chinese_TW.txt  French.txt  Japanese.txt  notices.txt  Slovenian.txt
Chinese.txt     German.txt  Korean.txt    Polish.txt    Spanish.txt
Czech.txt      Greek.txt   Lithuanian.txt Portuguese.txt Status.dat
English.txt     Indonesian.txt Italian.txt  non_ibm_license.txt Russian.txt  Turkish.txt
```

The license agreement remains available in the extraction target directory under the `license` subdirectory for future access. The license files are written using operating system-specific code pages. This enables you to view the license in English and in the local language configured on your machine. The other languages are not guaranteed to be viewable.

Verifying signature of IBM Spectrum Scale packages

All IBM Spectrum Scale packages for Red Hat Enterprise Linux and SLES operating systems on all supported architectures are signed with a GPG (GNU Privacy Guard) key by IBM. The repository metadata is also signed by IBM. You can verify that an IBM Spectrum Scale package and repository metadata are signed by IBM as follows.

The public key is located in a file that is called `SpectrumScale_public_key.pgp` and this file is present in the IBM Spectrum Scale installation images that can be downloaded from IBM Fix Central. For the latest version of the public key, see [IBM Spectrum Scale FAQ](#) in IBM Documentation.

Important: If you are using the installation toolkit, no additional steps are required. The installation toolkit checks the signature of each package and the repository metadata automatically before installation or upgrade.

For manual installation or upgrade, if you do not want to verify that the packages are signed, no additional steps are required. The signed packages function the same as the unsigned packages. If you want to manually verify that the packages are signed by IBM, do the following steps.

1. Import the public key into the RPM database.

```
rpm --import SpectrumScale_public_key.pgp
```

Note: Some Red Hat packages are required for object installation. These packages are signed by Red Hat with a GPG key. This key is called `RPM-GPG-KEY-redhat-release` and it is located in the `/usr/lpp/mmfs/<release>/Public_Keys/` directory. If you want to manually install object, along with importing the IBM public key, you can import the Red Hat key as follows.

```
rpm --import /usr/lpp/mmfs/<release>/Public_Keys/RPM-GPG-KEY-redhat-release
```

2. Confirm that the public key is imported into the RPM database.

```
rpm -q gpg-pubkey --qf '%{NAME}-%{VERSION}-%{RELEASE} \ n%{INSTALLTIME:date} \ n%{SUMMARY} \n\n' | grep SpectrumScale
```

3. Check the package's signature.

```
rpm -K PackageName
```

You can check the signature of more than one package by using wildcard characters. For example:

```
rpm -K *.rpm.
```

Extracting IBM Spectrum Scale patches (update SLES and Red Hat Enterprise Linux RPMs or Ubuntu Linux packages)

Typically, when you install an IBM Spectrum Scale system there are patches available. It is recommended that you always look for the latest patches when installing or updating an IBM Spectrum Scale node.

Note: For information about restrictions pertaining to Ubuntu Linux, see the [IBM Spectrum Scale FAQ in IBM Documentation](#).

IBM Spectrum Scale patches (update SLES and Red Hat Enterprise Linux RPMs and Ubuntu Linux packages) are available from [Fix Central](#).

The updated SLES and Red Hat Enterprise Linux RPMs and Ubuntu Linux packages are distributed as self-extracting base software.

If you are applying a patch during the initial installation of IBM Spectrum Scale on a node, you only need to build the portability layer once after the base and update SLES or Red Hat Enterprise Linux RPMs or Ubuntu Linux packages are installed.

Related tasks

[“Building the GPFS portability layer on Linux nodes” on page 364](#)

Before starting GPFS, you must build and install the GPFS portability layer.

Installing the IBM Spectrum Scale man pages on Linux nodes

To use the IBM Spectrum Scale man pages, the `gpfs.docs` package (rpm or deb package) must be installed.

Once you have installed the `gpfs.docs` package, the IBM Spectrum Scale man pages are located at `/usr/share/man/`.

You do not need to install the `gpfs.docs` package on all nodes if man pages are not desired (for example, if local file system space on the node is minimal).

If you are using the installation toolkit on supported Linux distributions, it automatically installs the associated man pages when it runs **spectrumscale install** to create a new GPFS cluster.

For Linux on Z: Changing the kernel settings

For IBM Spectrum Scale to run on Linux on Z, the option `"vmalloc=4096G"` needs to be added to the kernel settings.

Before starting IBM Spectrum Scale, perform the following steps on each Linux on Z node.

If you are using Red Hat Enterprise Linux 7:

1. Edit the `/etc/zipl.conf` file and add `vmalloc=4096G` as shown below:

```
cat /etc/zipl.conf
Parameters = "... vmalloc=4096G"
```

2. Run the **zipl** command.

```
zipl -V
```

3. Reboot the node.

If you are using Red Hat Enterprise Linux 8:

1. Use the `grubby` utility to add `vmalloc=4096G` for all boot entries on your node as shown in the following example:

```
grubby --update-kernel=ALL --args="vmalloc=4096G"
```

2. Run the **zipl** command.

```
zipl -V
```

3. Reboot the node.

If you are using SLES:

1. Edit the `/etc/default/grub` file and add the following:

```
cat/etc/default/grub
GRUB_CMDLINE_LINUX_DEFAULT=" ... vmalloc=4096G "
```

2. Run the **grub2-mkconfig** command.

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

3. Reboot the node.

Note: For more detailed information about installation and startup of IBM Spectrum Scale on IBM Z, see [Getting started with IBM Spectrum Scale for Linux on IBM Z](#).

Manually installing the IBM Spectrum Scale software packages on Linux nodes

Use this information to manually install the IBM Spectrum Scale software on Linux nodes.

Note: You can also use the installation toolkit to install and configure packages on supported Linux distributions. For more information, see [“Overview of the installation toolkit” on page 386](#).

You can install the IBM Spectrum Scale packages manually by using tools such as **rpm** or **yum** on Red Hat Enterprise Linux, **rpm** or **zypper** on SLES, and **dpkg** or **apt** on Ubuntu Linux.

Required packages for SLES and Red Hat Enterprise Linux

The following packages are required for IBM Spectrum Scale Standard Edition or IBM Spectrum Scale Data Access Edition on SLES and Red Hat Enterprise Linux:

- `gpfs.base*.rpm`
- `gpfs.gpl*.noarch.rpm`
- `gpfs.msg.en_US*.noarch.rpm`
- `gpfs.gskit*.rpm`
- `gpfs.license*.rpm`

The `gpfs.crypto*.rpm` and `gpfs.adv*.rpm` packages are required for following editions respective to the operating system, in addition to the packages listed for IBM Spectrum Scale Standard Edition or IBM Spectrum Scale Data Access Edition:

- IBM Spectrum Scale Advanced Edition and IBM Spectrum Scale Data Management Edition on SLES
- IBM Spectrum Scale Advanced Edition, IBM Spectrum Scale Data Management Edition, and IBM Spectrum Scale Developer Edition on Red Hat Enterprise Linux

Required packages for Ubuntu Linux

The following packages are required for IBM Spectrum Scale Standard Edition or IBM Spectrum Scale Data Access Edition on Ubuntu Linux:

- `gpfs.base*.deb`
- `gpfs.gpl*all.deb`
- `gpfs.msg.en-us*all.deb`
- `gpfs.gskit*.deb`
- `gpfs.license*.deb`

The following packages are required for IBM Spectrum Scale Advanced Edition and IBM Spectrum Scale Data Management Edition on Ubuntu Linux, in addition to the packages listed for IBM Spectrum Scale Standard Edition or IBM Spectrum Scale Data Access Edition:

- `gpfs.crypto*.deb`
- `gpfs.adv*.deb`

Note: For this release, the IBM Global Security Kit (GSKit) version for RPMs and Ubuntu Linux packages must be at least 8.0.55.x or later.

Optional packages for SLES and Red Hat Enterprise Linux

- `gpfs.docs*noarch.rpm`
- `gpfs.java*.rpm`
- `gpfs.compression*.rpm`
- `gpfs.gui*.rpm`
- `gpfs.librdkafka*.x86_64.rpm` (Red Hat Enterprise Linux x86_64 only)
- `gpfs.afm.cos*.rpm`

For information about manually installing management GUI packages, see [“Manually installing IBM Spectrum Scale management GUI”](#) on page 379.

The following performance monitoring packages are also optional for SLES and Red Hat Enterprise Linux.

- `gpfs.gss.pmcollector*.rpm`
- `gpfs.gss.pmsensors*.rpm`

The following protocols packages are available for Red Hat Enterprise Linux 8.x:

- `gpfs.nfs-ganesha*.el8.*.rpm`
- `gpfs.nfs-ganesha-gpfs*.el8.*.rpm`
- `gpfs.nfs-ganesha-utils*.el8.*.rpm`
- `gpfs.smb*_gpfs_*.el8.*.rpm`
- `spectrum-scale-object*.noarch.rpm`
- `gpfs.pm-ganesha*.el8.*.rpm`
- `pmswift*.noarch.rpm`

The following protocols packages are available for Red Hat Enterprise Linux 7.x:

- `gpfs.nfs-ganesha*.el7.*.rpm`
- `gpfs.nfs-ganesha-gpfs*.el7.*.rpm`
- `gpfs.nfs-ganesha-utils*.el7.*.rpm`
- `gpfs.smb*_gpfs_*.el7.*.rpm`
- `gpfs.pm-ganesha*.el7.*.rpm`

The following protocols packages are available for SLES 15:

- `gpfs.nfs-ganesha*.sles15.x86_64.rpm`
- `gpfs.nfs-ganesha-gpfs*.sles15.x86_64.rpm`
- `gpfs.nfs-ganesha-utils*.sles15.x86_64.rpm`
- `gpfs.smb*_gpfs_*.sles15.x86_64.rpm`
- `gpfs.pm-ganesha*.sles15.x86_64.rpm`

For information on manually installing packages including protocols packages on Red Hat Enterprise Linux, SLES, and Ubuntu nodes, see [“Manually installing IBM Spectrum Scale and deploying protocols on Linux nodes”](#) on page 366.

For information about manually installing performance monitoring packages, see [“Manually installing the performance monitoring tool” on page 376](#).

Optional packages for Ubuntu Linux

The following packages are optional for Ubuntu Linux.

- `gpfs.docs*all.deb`
- `gpfs.librdkafka*.deb`
- `gpfs.afm.cos*.deb`
- `gpfs.compression*.deb`

The following performance monitoring packages are optional for Ubuntu Linux (x86_64 and s390x).

- `gpfs.gss.pmcollector*20*.deb`
- `gpfs.gss.pmsensors*20*.deb`

The following protocols packages are available for Ubuntu 20.04:

- `gpfs.nfs-ganesha*~focal_amd64.deb`
- `gpfs.nfs-ganesha_gpfs*~focal_amd64.deb`
- `gpfs.nfs-ganesha-dbgsym*~focal_amd64.deb`
- `gpfs.python-nfs-ganesha*~focal_all.deb`
- `gpfs.nfs-ganesha-doc*all.~focal_deb`
- `gpfs.smb*gpfs*~focal_amd64.deb`
- `gpfs.smb-dbg_*gpfs*~focal_amd64.deb`
- `gpfs.pm-ganesha*~focal_amd64.deb`

The following protocols packages are available for Ubuntu 22.04:

- `gpfs.nfs-ganesha*~jammy_amd64.deb`
- `gpfs.nfs-ganesha_gpfs*~jammy_amd64.deb`
- `gpfs.nfs-ganesha-dbgsym*~jammy_amd64.deb`
- `gpfs.python-nfs-ganesha*~jammy_all.deb`
- `gpfs.nfs-ganesha-doc*all.~jammy_deb`
- `gpfs.smb*gpfs*~jammy_amd64.deb`
- `gpfs.smb-dbg_*gpfs*~jammy_amd64.deb`
- `gpfs.pm-ganesha*~jammy_amd64.deb`

The following packages are required (and provided) only on the Elastic Storage Server (ESS):

- `gpfs.gnr.base*.ppc64.rpm`
- `gpfs.gnr*.ppc64.rpm`
- `gpfs.gss.firmware*.ppc64.rpm`

For more information about Elastic Storage Server (ESS), see [Elastic Storage Server \(ESS\) documentation](#).

Location of extracted packages

The installation images are extracted to following component specific directories.

GPFS packages

- Red Hat Enterprise Linux: `/usr/lpp/mmfs/5.1.x.x/gpfs_rpms`
 - File audit logging and watch folder packages: `/usr/lpp/mmfs/5.1.x.x/gpfs_rpms/rhel`
- Ubuntu: `/usr/lpp/mmfs/5.1.x.x/gpfs_debs`
 - File audit logging and watch folder packages: `/usr/lpp/mmfs/5.1.x.x/gpfs_debs/ubuntu`

- SLES: /usr/lpp/mmfs/5.1.x.x/gpfs_rpms

NFS packages

- Red Hat Enterprise Linux 8.x: /usr/lpp/mmfs/5.1.x.x/ganesha_rpms/rhel8
- Red Hat Enterprise Linux 7.x: /usr/lpp/mmfs/5.1.x.x/ganesha_rpms/rhel7
- Ubuntu 20.04: /usr/lpp/mmfs/5.1.x.x/ganesha_debs/ubuntu/ubuntu20
- Ubuntu 22.04: /usr/lpp/mmfs/5.1.x.x/ganesha_debs/ubuntu/ubuntu22
- SLES 15: /usr/lpp/mmfs/5.1.x.x/ganesha_rpms/sles15

SMB packages

- Red Hat Enterprise Linux 8.x: /usr/lpp/mmfs/5.1.x.x/smb_rpms/rhel8
- Red Hat Enterprise Linux 7.x: /usr/lpp/mmfs/5.1.x.x/smb_rpms/rhel7
- Ubuntu 20.04: /usr/lpp/mmfs/5.1.x.x/smb_debs/ubuntu/ubuntu20
- Ubuntu 22.04: /usr/lpp/mmfs/5.1.x.x/smb_debs/ubuntu/ubuntu22
- SLES 15: /usr/lpp/mmfs/5.1.x.x/ganesha_rpms/sles15

Object packages

- Red Hat Enterprise Linux 8.x: /usr/lpp/mmfs/5.1.x.x/object_rpms/rhel8

Performance monitoring packages

- Red Hat Enterprise Linux 8.x: /usr/lpp/mmfs/5.1.x.x/zimon_rpms/rhel8
- Red Hat Enterprise Linux 7.x: /usr/lpp/mmfs/5.1.x.x/zimon_rpms/rhel7
- Ubuntu 20.04: /usr/lpp/mmfs/5.1.x.x/zimon_debs/zimon/ubuntu20
- Ubuntu 22.04: /usr/lpp/mmfs/5.1.x.x/zimon_debs/zimon/ubuntu22
- SLES 15: /usr/lpp/mmfs/5.1.x.x/zimon_rpms/sles15

Note: The pm-ganesha package is extracted in the following directories:

- Red Hat Enterprise Linux 8.x: /usr/lpp/mmfs/5.1.x.x/zimon_rpms/rhel8
- **Note:** The pmswift package is also extracted on Red Hat Enterprise Linux 8.x.
- Red Hat Enterprise Linux 7.x: /usr/lpp/mmfs/5.1.x.x/zimon_rpms/rhel7
- SLES: /usr/lpp/mmfs/5.1.x.x/zimon_rpms/sles15
- Ubuntu: /usr/lpp/mmfs/5.1.x.x/zimon_debs/ubuntu

Installation of packages for SLES or Red Hat Enterprise Linux

To install all of the required GPFS SLES or Red Hat Enterprise Linux RPMs for the IBM Spectrum Scale Standard Edition or IBM Spectrum Scale Data Access Edition, change the directory to where the installation image is extracted. For example, issue this command:

```
cd /usr/lpp/mmfs/5.1.5.x/gpfs_rpms
```

Then, issue this command for Standard Edition:

```
rpm -ivh gpfs.base*.rpm gpfs.gpl*.rpm gpfs.license.std*.rpm gpfs.gskit*.rpm gpfs.msg*.rpm  
gpfs.docs*.rpm
```

or issue the following command for Data Access Edition:

```
rpm -ivh gpfs.base*.rpm gpfs.gpl*.rpm gpfs.license.da*.rpm gpfs.gskit*.rpm gpfs.msg*.rpm  
gpfs.docs*.rpm
```

To install all of the required GPFS SLES or Red Hat Enterprise Linux RPMs for the IBM Spectrum Scale Advanced Edition, IBM Spectrum Scale Data Management Edition, or IBM Spectrum Scale Developer

Edition, change to the directory where the installation image is extracted. For example, enter the following command:

```
cd /usr/lpp/mmfs/5.1.5.x/gpfs_rpms
```

Then, issue the following command for Advanced Edition:

```
rpm -ivh gpfs.base*.rpm gpfs.gpl*.rpm gpfs.license.adv*.rpm gpfs.gskit*.rpm  
gpfs.docs*.rpm gpfs.msg*.rpm gpfs.adv*.rpm gpfs.crypto*.rpm
```

or issue the following command for Data Management Edition:

```
rpm -ivh gpfs.base*.rpm gpfs.gpl*.rpm gpfs.license.dm*.rpm gpfs.gskit*.rpm  
gpfs.docs*.rpm gpfs.msg*.rpm gpfs.adv*.rpm gpfs.crypto*.rpm
```

or issue the following command for Developer Edition:

```
rpm -ivh gpfs.base*.rpm gpfs.gpl*.rpm gpfs.license.dev*.rpm gpfs.gskit*.rpm  
gpfs.docs*.rpm gpfs.msg*.rpm gpfs.adv*.rpm gpfs.crypto*.rpm
```

Note: IBM Spectrum Scale Developer Edition is available only on Red Hat Enterprise Linux on x86_64.

Installation of packages for Ubuntu Linux

To install all of the GPFS Ubuntu Linux packages for the IBM Spectrum Scale Standard edition, change the directory to where the installation image is extracted. For example, issue this command:

```
cd /usr/lpp/mmfs/5.1.5.x/gpfs_debs
```

then, issue this command for Standard Edition:

```
apt install ./gpfs.base*deb ./gpfs.gpl*deb ./gpfs.license.std*.deb ./gpfs.gskit*deb ./  
gpfs.msg*deb ./gpfs.docs*deb
```

or issue the following command for Data Access Edition:

```
apt install ./gpfs.base*deb ./gpfs.gpl*deb ./gpfs.license.da*.deb ./gpfs.gskit*deb ./  
gpfs.msg*deb ./gpfs.docs*deb
```

To install all of the GPFS Ubuntu Linux packages for IBM Spectrum Scale Advanced Edition and IBM Spectrum Scale Data Management Edition, change the directory to where the installation image is extracted. For example, issue this command:

```
cd /usr/lpp/mmfs/5.1.5.x/gpfs_debs
```

then, issue this command for Advanced Edition:

```
apt install ./gpfs.base*deb ./gpfs.gpl*deb ./gpfs.license.adv*.deb ./gpfs.gskit*deb ./  
gpfs.msg*deb ./gpfs.docs*deb ./gpfs.adv*deb ./gpfs.crypto*deb
```

or, issue this command for Data Management Edition:

```
apt install ./gpfs.base*deb ./gpfs.gpl*deb ./gpfs.license.dm*.deb ./gpfs.gskit*deb ./  
gpfs.msg*deb ./gpfs.docs*deb ./gpfs.adv*deb ./gpfs.crypto*deb
```

Related concepts

[“Manually installing IBM Spectrum Scale management GUI” on page 379](#)

The management GUI provides an easy way for the users to configure, manage, and monitor the IBM Spectrum Scale system.

[“Installing call home” on page 487](#)

The call home feature collects files, logs, traces, and details of certain system health events from different nodes and services. These details are shared with the IBM support center for monitoring and problem determination.

[“Manually installing the performance monitoring tool” on page 376](#)

The performance monitoring tool is automatically installed by the installation toolkit. You can also install the performance monitoring tool manually.

[“Object protocol further configuration” on page 448](#)

If the Object protocol is enabled during installation, use the following information to verify the Object protocol configuration. You can also enable features such as unified file and object access and multi-region object deployment.

Building the GPFS portability layer on Linux nodes

Before starting GPFS, you must build and install the GPFS portability layer.

The GPFS portability layer is a loadable kernel module that allows the GPFS daemon to interact with the operating system.

Note: The GPFS kernel module should be updated every time the Linux kernel is updated. Updating the GPFS kernel module after a Linux kernel update requires rebuilding and installing a new version of the module.

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 Spectrum 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 Spectrum Scale: Command and Programming Reference*.

To build the GPFS portability layer on Linux nodes, do the following:

1. Check for the following before building the portability layer:
 - Updates to the portability layer at the IBM Support Portal: Downloads for General Parallel File System (www.ibm.com/support/entry/portal/Downloads/Software/Cluster_software/General_Parallel_File_System).
 - The latest kernel levels supported in the *IBM Spectrum Scale FAQ* in *IBM Documentation*.
 - The kernel development files and compiler utilities that are required to build the portability layer in [“Software requirements” on page 220](#).
2. Build your GPFS portability layer in one of the following ways:
 - Using the `mmbuildgpl` command (recommended) for nodes running GPFS 4.1.0.4 or later. For more information, see *mmbuildgpl command* in *IBM Spectrum Scale: Command and Programming Reference*.
 - Using the Autoconfig tool.
 - Using the directions in `/usr/lpp/mmfs/src/README`.

Using the mmbuildgpl command to build the GPFS portability layer on Linux nodes

Starting with GPFS 4.1.0.4, you can use the `mmbuildgpl` command to simplify the build process.

To build the GPFS portability layer using `mmbuildgpl`, enter the following command:

```
/usr/lpp/mmfs/bin/mmbuildgpl
```

Each kernel module is specific to a Linux version and platform. If you have multiple nodes running exactly the same operating system level on the same platform, and only some of these nodes have a compiler available, you can build the kernel module on one node, then create an installable package that contains the binary module for ease of distribution.

If you choose to generate an installable package for portability layer binaries, perform the following additional step:

```
/usr/lpp/mmfs/bin/mmbuildgpl --build-package
```

When the command finishes, it displays the location of the generated package as in the following examples:

```
Wrote: /root/rpmbuild/RPMS/x86_64/gpfs.gplbin-4.18.0-305.12.1.el8_4.x86_64-5.1.5-x.x86_64.rpm
```

or

```
Wrote: /tmp/deb/gpfs.gplbin-5.4.0-81-generic_5.1.5-x_amd64.deb
```

You can then copy the generated package to other machines for deployment. By default, the generated package can be deployed only to machines whose architecture, distribution level, Linux kernel, and IBM Spectrum Scale maintenance level are identical with those of the machine on which the `gpfs.gplbin` package was built. However, you can install the generated package on a machine with a different Linux kernel by setting the `MM_INSTALL_ONLY` environment variable before you install the generated package. If you install the `gpfs.gplbin` package, you do not need to install the `gpfs.gpl` package.

Note: During the package generation, temporary files are written to the `/tmp/rpm` or `/tmp/deb` directory, so be sure there is sufficient space available. By default, the generated package goes to `/usr/src/packages/RPMS/<arch>` for SUSE Linux Enterprise Server, `/usr/src/redhat/RPMS/<arch>` for Red Hat Enterprise Linux, and `/tmp/deb` for Ubuntu Linux.

Important:

- The GPFS portability layer is specific to both the current kernel and the GPFS version. If either the kernel or the GPFS version changes, a new GPFS portability layer needs to be built.
- Although operating system kernels might upgrade to a new version, they are not active until after a reboot. Thus, a GPFS portability layer for this new kernel must be built after a reboot of the operating system.
- Before you install a new GPFS portability layer, make sure to uninstall the prior version of the GPFS portability layer first.

Using the Autoconfig tool to build the GPFS portability layer on Linux nodes

To simplify the build process, GPFS provides an automatic configuration tool called `Autoconfig`.

The following example shows the commands required to build the GPFS portability layer using the `Autoconfig` tool:

```
cd /usr/lpp/mmfs/src
make Autoconfig
make World
make InstallImages
```

Each kernel module is specific to a Linux version and platform. If you have multiple nodes running exactly the same operating system level on the same platform, or if you have a compiler available on only one node, you can build the kernel module on one node, then create an installable package that contains the binary module for ease of distribution.

If you choose to generate an installable package for portability layer binaries, perform the following additional step:

make rpm for SLES and RHEL Linux

make deb for Ubuntu Linux

When the command finishes, it displays the location of the generated package as in the following examples:

```
Wrote: /root/rpmbuild/RPMS/x86_64/gpfs.gplbin-4.18.0-305.12.1.el8_4.x86_64-5.1.5-x.x86_64.rpm
```

or

```
Wrote: /tmp/deb/gpfs.gplbin-5.4.0-81-generic_5.1.5-x_amd64.deb
```

You can then copy the generated package to other machines for deployment. By default, the generated package can be deployed only to machines whose architecture, distribution level, Linux kernel, and IBM Spectrum Scale maintenance level are identical with those of the machine on which the `gpfs.gplbin` package was built. However, you can install the generated package on a machine with a different Linux kernel by setting the `MM_INSTALL_ONLY` environment variable before you install the generated package. If you install the `gpfs.gplbin` package, you do not need to install the `gpfs.gpl` package.

Note: During the package generation, temporary files are written to the `/tmp/rpm` or `/tmp/deb` directory, so be sure there is sufficient space available. By default, the generated package goes to `/usr/src/packages/RPMS/<arch>` for SUSE Linux Enterprise Server, `/usr/src/redhat/RPMS/<arch>` for Red Hat Enterprise Linux, and `/tmp/deb` for Ubuntu Linux.

Important:

- The GPFS portability layer is specific to both the current kernel and the GPFS version. If either the kernel or the GPFS version changes, a new GPFS portability layer needs to be built.
- Although operating system kernels might upgrade to a new version, they are not active until after a reboot. Thus, a GPFS portability layer for this new kernel must be built after a reboot of the operating system.
- Before you install a new GPFS portability layer, make sure to uninstall the prior version of the GPFS portability layer first.

Manually installing IBM Spectrum Scale and deploying protocols on Linux nodes

The following tasks describe how to manually install IBM Spectrum Scale and deploy protocols on systems running on supported Linux distributions.

Note: CES supports HDFS protocols. CES HDFS follows the same installation methods and prerequisites as the other protocols as documented in this section. For specific information about installing CES HDFS, see *Installation* under [CES HDFS in Big data and analytics support documentation](#).

These prerequisites must be met before installing IBM Spectrum Scale on Linux systems.

- Repository must be set up.
 - For information on setting up repository on Red Hat Enterprise Linux, see *Red Hat Enterprise Linux System Administrator's Guide*.
 - For information on setting up repository on SLES, see *SLES Deployment Guide*.
 - For information on setting up repository on Ubuntu, see *Ubuntu Server Guide*.
- All prerequisite packages must be installed. For a list of prerequisite packages, see [“Software requirements”](#) on page 220 and [“Installation prerequisites”](#) on page 352.
- Passwordless SSH must be set up between all nodes in the cluster. For more information, see *Problems due to missing prerequisites* in *IBM Spectrum Scale: Problem Determination Guide*.
- Firewall configuration must be according to your requirements. It is recommended that firewalls are in place to secure all nodes. For more information, see *Securing the IBM Spectrum Scale system using firewall* in *IBM Spectrum Scale: Administration Guide*.

To check the status of the firewall on Red Hat Enterprise Linux, issue the following command:

```
systemctl status firewalld
```

To check the status of the firewall on SLES, issue the following command:

```
sudo /sbin/rcSuSEfirewall2 status
```

To check the status of the firewall on Ubuntu, issue the following command:

```
sudo ufw status
```

For examples of how to open firewall ports on different operating systems, see *Examples of how to open firewall ports* in *IBM Spectrum Scale: Administration Guide*.

- Every node must have a non-loopback IP address assigned. In some scenarios, a freshly installed node might have its host name pointing to 127.0.0.1 in `/etc/hosts`. 127.0.0.1 is a loopback IP address and it is not sufficient for multi-node IBM Spectrum Scale cluster creation. In these cases, each node needs a static IP with connectivity to the other nodes.
- Optionally, the bash profile can be updated to allow easier access to IBM Spectrum Scale commands.
 - Verify that the PATH environment variable for the root user on each node includes `/usr/lpp/mmfs/bin` and the required tool directories. This allows a user to execute IBM Spectrum Scale commands without having to first change directory to `/usr/lpp/mmfs/bin`.
 - Export the `WCOLL` variable used by **mmdsh**. In the following example, `/nodes` is a manually created file, listing line by line, each node within the cluster using the nodes' FQDN. Once IBM Spectrum Scale is installed, this configuration allows the **mmdsh** command to execute commands on multiple nodes simultaneously.

Example:

```
# cat ~/.bash_profile
# .bash_profile
# Get the aliases and functions
if [ -f ~/.bashrc ]; then
. ~/.bashrc
fi

#####
# User specific environment and startup programs
#####
#####
# Specifics for GPFS testing
#####
export PATH=$PATH:$HOME/bin:/usr/lpp/mmfs/bin
export WCOLL=/nodes
```

Log out and then log in again for the changes in the bash profile to take effect.

Installing IBM Spectrum Scale packages on Linux systems

Manually install IBM Spectrum Scale packages on systems that are running on supported Linux distributions as follows.

Do these steps on all nodes on which you want to install IBM Spectrum Scale, one by one unless noted otherwise.

1. Download and extract IBM Spectrum Scale packages, and then accept the licensing agreement. For more information, see [“Extracting the IBM Spectrum Scale software on Linux nodes”](#) on page 355 and [“Accepting the electronic license agreement on Linux nodes”](#) on page 355.

For information on the location of extracted packages, see [“Location of extracted packages”](#) on page 361.

2. Install the IBM Spectrum Scale packages by issuing one of the following commands depending on the operating system.

Note: The following command example shows the installation of IBM Spectrum Scale Advanced Edition packages.

- On Red Hat Enterprise Linux, and SLES, issue the following command.

```
# cd /usr/lpp/mmfs/x.x.x.x/gpfs_rpms
# rpm -ivh gpfs.base*.rpm gpfs.gpl*.rpm gpfs.license*.rpm gpfs.gskit*.rpm
gpfs.msg*.rpm gpfs.compression*.rpm gpfs.adv*.rpm gpfs.crypto*.rpm
```

- On Ubuntu, issue the following command.

```
# cd /usr/lpp/mmfs/x.x.x.x/gpfs_debs
# dpkg -i gpfs.base*.deb gpfs.gpl*.deb gpfs.license.ad*.deb gpfs.gskit*.deb
gpfs.msg*.deb gpfs.compression*.deb gpfs.adv*.deb gpfs.crypto*.deb
```

A sample output is as follows.

```
Preparing... ##### [100%]
1:gpfs.base ##### [ 11%]
2:gpfs.compression ##### [ 27%]
3:gpfs.adv ##### [ 33%]
4:gpfs.crypto ##### [ 44%]
5:gpfs.gpl ##### [ 56%]
6:gpfs.license.adv ##### [ 67%]
7:gpfs.msg.en_US ##### [ 83%]
8:gpfs.gskit ##### [ 100%]
```

- Build the portability layer by using the following command.

```
# /usr/lpp/mmfs/bin/mmbuildgpl
```

A sample output is as follows. The output varies depending on the operating system.

```
-----
mmbuildgpl: Building GPL module begins at Thu Mar 24 15:18:23 CST 2016.
-----
Verifying Kernel Header...
kernel version = 31228004 (3.12.28-4-default, 3.12.28-4)
module include dir = /lib/modules/3.12.28-4-default/build/include
module build dir = /lib/modules/3.12.28-4-default/build
kernel source dir = /usr/src/linux-3.12.28-4/include
Found valid kernel header file under /lib/modules/3.12.28-4-default/build/include
Verifying Compiler...
make is present at /usr/bin/make
cpp is present at /usr/bin/cpp
gcc is present at /usr/bin/gcc
g++ is present at /usr/bin/g++
ld is present at /usr/bin/ld
Verifying Additional System Headers...
Verifying linux-glibc-devel is installed ...
Command: /bin/rpm -q linux-glibc-devel
The required package linux-glibc-devel is installed
make World ...
make InstallImages ...
-----
mmbuildgpl: Building GPL module completed successfully at Thu Mar 24 15:18:31 CST 2016.
-----
```

If the portability layer cannot be built due to missing prerequisite packages, you might need to install prerequisite packages. For a list of prerequisite packages, see [“Software requirements”](#) on page 220.

If the repository setup and the kernel configuration are correct, you can install the prerequisite packages by using one of the following commands, depending on the operating system:

Red Hat Enterprise Linux

```
# /usr/bin/yum install -y package_name1 package_name2 ... package_nameN
```

SLES

```
# /usr/bin/zypper install -y package_name1 package_name2 ... package_nameN
```

Ubuntu

```
# /usr/bin/apt-get install -y package_name1 package_name2 ... package_nameN
```


After IBM Spectrum Scale is built on all nodes, you can create the cluster. It is recommended to have an odd number of quorum nodes and that your NSD nodes be designated as quorum nodes.

4. Create the cluster by using the following command from one of the nodes.

```
# /usr/lpp/mmfs/bin/mmcrcluster -N NodesList --ccr-enable -r /usr/bin/ssh -R /usr/bin/scp -C cluster1.spectrum
```

In this command example, *NodesList* is a file that contains a list of nodes and node designations to be added to the cluster and its contents are as follows:

```
node1:quorum
node2
node3
node4:quorum-manager
node5:quorum-manager
```

Running the **mmcrcluster** command generates output similar to the following snippet:

```
mmcrcluster: Performing preliminary node verification ...
mmcrcluster: Processing quorum and other critical nodes ...
mmcrcluster: Processing the rest of the nodes ...
mmcrcluster: Finalizing the cluster data structures ...
mmcrcluster: Command successfully completed
mmcrcluster: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
# Thu Mar 24 15:33:06 CST 2016: mmcommon pushSdr_async: mmsdrfs propagation started
Thu Mar 24 15:33:10 CST 2016: mmcommon pushSdr_async: mmsdrfs propagation completed; mmdsh
rc=0
```

5. Accept proper role licenses for the nodes by using the following commands from one of the nodes.

- a) Accept the server licenses for the applicable nodes.

```
# /usr/lpp/mmfs/bin/mmchlicense server --accept -N node1,node4,node5
```

A sample output is as follows:

```
The following nodes will be designated as possessing server licenses:
node1
node4
node5
mmchlicense: Command successfully completed
mmchlicense: Warning: Not all nodes have proper GPFS license designations.
Use the mmchlicense command to designate licenses as needed.
mmchlicense: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
# Thu Mar 24 15:37:59 CST 2016: mmcommon pushSdr_async: mmsdrfs propagation started
Thu Mar 24 15:38:01 CST 2016: mmcommon pushSdr_async: mmsdrfs propagation completed;
mmdsh rc=0
```

- b) Accept the client licenses for the applicable nodes.

```
# /usr/lpp/mmfs/bin/mmchlicense client --accept -N node2,node3
```

A sample output is as follows:

```
The following nodes will be designated as possessing client licenses:
node2
node3
mmchlicense: Command successfully completed
mmchlicense: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
# Thu Mar 24 15:38:26 CST 2016: mmcommon pushSdr_async: mmsdrfs propagation started
Thu Mar 24 15:38:28 CST 2016: mmcommon pushSdr_async: mmsdrfs propagation completed;
mmdsh rc=0
```

IBM Spectrum Scale cluster is now created. You can view the configuration information of the cluster by using the following command.

```
# /usr/lpp/mmfs/bin/mmlscluster
```

The system displays output similar to the following snippet:

```
GPFS cluster information
=====
GPFS cluster name:      cluster1.spectrum
GPFS cluster id:        993377111835434248
GPFS UID domain:        cluster1.spectrum
Remote shell command:   /usr/bin/ssh
Remote file copy command: /usr/bin/scp
Repository type:        CCR
GPFS cluster configuration servers:
-----
Primary server:         node1 (not in use)
Secondary server:       (none)
Node Daemon node name IP address Admin node name Designation
-----
1 node1 203.0.113.11 node1 quorum
2 node4 203.0.113.14 node4 quorum-manager
3 node5 203.0.113.15 node5 quorum-manager
4 node2 203.0.113.12 node2
5 node3 203.0.113.13 node3
```

6. Start the GPFS daemons and the cluster by using the following command from one of the nodes.

```
# /usr/lpp/mmfs/bin/mmstartup -N NodesList
```

Where *NodesList* is the list of nodes on which the daemons and the cluster are to be started.

You can use the **mmgetstate -N NodesList** command to verify that the GPFS software is running on these nodes.

Creating NSDs and file systems as part of installing IBM Spectrum Scale on Linux systems

Create NSDS and file systems for installing IBM Spectrum Scale on supported Linux distributions as follows.

For information about NSD creation considerations, see [“Network Shared Disk \(NSD\) creation considerations”](#) on page 239.

1. Create NSDs as follows.

- a) Identify available physical disks in the `/dev` directory.
- b) Create the NSD stanza file to be used with the **mmcrnsd** command.

For example, consider you have 2 disks, `/dev/sdc` and `/dev/sdd`, and your node name is `exnode1` and it is running Linux. In this case, your NSD stanza file contents are similar to the following:

```
%nsd:
    device=/dev/sdc
    nsd=nsd1
    servers=exnode1
    usage=dataAndMetadata
    failureGroup=-1
    pool=system
%nsd:
    device=/dev/sdd
    nsd=nsd2
    servers=exnode1
    usage=dataAndMetadata
    failureGroup=-1
```

Note: The server name used in the NSD stanza file must be resolvable by the system.

- c) Create the NSDs using the following command.

```
# mmcrnsd -F NSD_Stanza_Filename
```

2. Create a GPFS file system using the following command.

```
# mmcrfs fs1 -F NSD_Stanza_Filename -k nfs4
```



Attention: When creating NSD stanza files ensure that you plan out the required file system layout. This includes number of file systems, pools in each file system, failure group layout, metadata versus data usage types, as well as other `mmcrnsd` and `mmcrnsd stanzafile` parameters. Additionally, a stanzafile created for NSD creation by using the `mmcrnsd` command, can also be used for file system creation through the `mmcrfs` command. If you are using an NSD stanzafile for file system creation then you must be aware that a single file system is created. If you require multiple file systems, then you must create a separate stanzafile for each file system.

Installing Cluster Export Services as part of installing IBM Spectrum Scale on Linux systems

Install Cluster Export Services (CES) on supported Linux distributions as follows.

Note:

- The object protocol is not supported on Red Hat Enterprise Linux 7.x.
- The object protocol is not supported on SLES and Ubuntu.
- The object protocol is not supported on the s390x architecture.

Important: If you plan to install and configure CES, all protocols available on your platform must be installed. Installing a subset of available protocols is not supported.

1. Unmount GPFS file systems and stop GPFS on all nodes by using the following command.

```
# mmshutdown -a
```

2. Configure the CES shared root file system using the following command.

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

Note: It is recommended to configure an independent file system as the CES shared root file system.

3. Start GPFS on all nodes in the cluster by using the following command.

```
# mmstartup -a
```

4. Enable CES on the required nodes by using the following command.

```
# mmchnode --ces-enable -N pnode1,pnode2,pnode3
```

5. Add to the protocol nodes the IP addresses designated to be used for CES connectivity by using the following command.

```
# mmces address add --ces-node pnode1 --ces-ip CESIPAddress
```

6. Verify the CES configuration by using the following commands.

```
# mmlscluster --ces  
# mmces address list
```

Currently, there are no services that are enabled. You can verify by using the `mmces service list -a` command. A sample output is as follows.

```
No CES services are enabled.
```

7. Download and extract IBM Spectrum Scale protocol packages, and then accept the licensing agreement. For more information, see [“Extracting the IBM Spectrum Scale software on Linux nodes” on page 355](#) and [“Accepting the electronic license agreement on Linux nodes” on page 355](#).

For information on the location of extracted packages, see [“Location of extracted packages” on page 361](#).

8. Remove conflicting Samba packages that might have been installed on each CES node by issuing one of the following commands, depending on the operating system.

- On Red Hat Enterprise Linux and SLES, issue the following command.

```
# mmdsh -N all rpm -e samba-common --nodeps
# mmdsh -N all rpm -e samba-client
```

- On Ubuntu, issue the following command.

```
# mmdsh -N all dpkg -P samba-common
# mmdsh -N all dpkg -P samba-client
```

9. Install IBM Spectrum Scale SMB package by issuing one of the following commands, depending on the operating system.

- On Red Hat Enterprise Linux and SLES, issue the following command.

```
# rpm -ivh gpfs.smb*.rpm
```

- On Ubuntu, issue the following command.

```
# dpkg -i gpfs.smb*.deb
```

For a list of packages for the current IBM Spectrum Scale release, see [“Manually installing the IBM Spectrum Scale software packages on Linux nodes”](#) on page 359.

10. Install IBM Spectrum Scale NFS packages by issuing one of the following commands, depending on the operating system.

- On Red Hat Enterprise Linux and SLES, issue the following command.

```
# rpm -ivh NFS_Package_Name1 NFS_Package_Name2 ... NFS_Package_NameN
```

- On Ubuntu, issue the following command.

```
# dpkg -i NFS_Package_Name1 NFS_Package_Name2 ... NFS_Package_NameN
```

If you cannot install NFS package because of any error by using the **dpkg -i** command, issue the following command:

```
# apt install ./NFS_Package_Name1 ./NFS_Package_Name2 ... ./NFS_Package_NameN
```

This command installs the specified local package files.

Note: Before installing IBM Spectrum Scale 5.1.x on Ubuntu operating system, make sure that upstream NFS Ganesha package is not installed or configured in the protocol nodes. If the NFS Ganesha package is installed or configured, then uninstall the package by using the **apt purge nfs-ganesha** command.

For a list of packages for the current IBM Spectrum Scale release, see [“Manually installing the IBM Spectrum Scale software packages on Linux nodes”](#) on page 359.

11. Install IBM Spectrum Scale for object storage package by using the following procedure.

- [“Manually installing IBM Spectrum Scale for object storage on Red Hat Enterprise Linux”](#) on page 374.

If you plan to use IPv6 addresses, you must enable the CES interface mode. For more information, see *Configuring CES protocol service IP addresses* in *IBM Spectrum Scale: Administration Guide*.

Enabling NFS, SMB, HDFS, and object on Linux systems

Use this information to enable NFS, SMB, HDFS, and object on Linux systems after installing the packages.

Before you begin enabling NFS, SMB, HDFS, and object, you must have installed the corresponding IBM Spectrum Scale packages.

For HDFS, follow the instructions in [Enable and Configure CES HDFS](#) under *IBM Spectrum Scale support for Hadoop in Big data and analytics support*.

- Enable NFS as follows.
 - a) If the kernel NFS service is running, mask and stop the kernel NFS service on all nodes where CES NFS needs to be installed using the following commands.

```
# systemctl mask nfs-server.service
# systemctl daemon-reload
# systemctl stop nfs
```

- b) Enable the NFS services using the following command.

```
# mmces service enable NFS
```

If you get the `/sbin/rpc.statd: not found [No such file or directory]` error, try the following workaround.

Run the following commands on every protocol node.

```
# ln -s /usr/sbin/rpc.statd /sbin/rpc.statd
# ln -s /sbin/rpcinfo /usr/sbin/rpcinfo
```

After you have run these commands, try enabling the NFS services again.

- Enable the SMB services using the following command.

```
# mmces service enable SMB
```

- Enable object on Red Hat Enterprise Linux nodes. For more information, see [“Manually installing IBM Spectrum Scale for object storage on Red Hat Enterprise Linux ”](#) on page 374.

Verifying the IBM Spectrum Scale installation on Ubuntu Linux nodes

You can verify the installation of the IBM Spectrum Scale Ubuntu Linux packages on each node.

To check that the software is successfully installed, use the `dpkg` command:

```
dpkg -l | grep gpfs
```

The system returns output similar to the following if you have IBM Spectrum Scale Standard Edition or IBM Spectrum Scale Data Access Edition installed:

```
ii gpfs.base          5.1.5-x      amd64  GPFS File Manager
ii gpfs.compression  5.1.5-x      amd64  IBM Spectrum Scale Compression Libraries
ii gpfs.docs         5.1.5-x      all    GPFS Server Manpages and Documentation
ii gpfs.gpl          5.1.5-x      all    GPFS Open Source Modules
ii gpfs.gskit        8.0.55.x     amd64  GPFS GSKit Cryptography Runtime
ii gpfs.librdkafka    5.1.5-x      amd64  librdkafka shared library installation
ii gpfs.msg.en-us     5.1.5-x      all    GPFS Server Messages - U.S. English
ii gpfs.license.std   5.1.5-x      amd64  IBM Spectrum Scale Standard Edition ITLM files
```

or

```
ii gpfs.license.da    5.1.5-x      amd64  IBM Spectrum Scale Data Access Edition ITLM files
```

If you have IBM Spectrum Scale Advanced Edition or IBM Spectrum Scale Data Management Edition installed, you should also see the following lines in the output:

```
ii gpfs.adv          5.1.5-x      amd64  GPFS Advanced Features
ii gpfs.crypto        5.1.5-x      amd64  GPFS Cryptographic Subsystem
ii gpfs.license.adv   5.1.5-x      amd64  IBM Spectrum Scale Advanced Edition ITLM files
```

or

```
ii gpfs.license.dm    5.1.5-x      amd64  IBM Spectrum Scale Data Management Edition ITLM files
```

Verifying the IBM Spectrum Scale installation on SLES and Red Hat Enterprise Linux nodes

You can verify the installation of the IBM Spectrum Scale SLES or Red Hat Enterprise Linux RPMs on each node.

To check that the software is successfully installed, use the `rpm` command:

```
rpm -qa | grep gpfs
```

The system returns output similar to the following if you have IBM Spectrum Scale Standard Edition or IBM Spectrum Scale Data Access Edition installed:

```
gpfs.docs-5.1.5-x
gpfs.base-5.1.5-x
gpfs.msg.en_US-5.1.5-x
gpfs.compression-5.1.5-x
gpfs.gpl-5.1.5-x
gpfs.gskit-8.0.55.x
gpfs.license.std-5.1.5-x

or

gpfs.license.da-5.1.5-x
```

If you have IBM Spectrum Scale Advanced Edition, IBM Spectrum Scale Data Management Edition, or IBM Spectrum Scale Developer Edition installed, you should also see the following in the output:

```
gpfs.crypto-5.1.5-x
gpfs.adv-5.1.5-x
gpfs.license.dm-5.1.5-x or gpfs.license.adv-5.1.5-x or gpfs.license.dev-5.1.5-x
```

Note: IBM Spectrum Scale Developer Edition is available on Red Hat Enterprise Linux on x86_64.

For installations that include IBM Spectrum Scale RAID, you should also see the following in the output:

```
gpfs.gnr-5.1.5-x
```

For more information about IBM Spectrum Scale RAID, see *IBM Spectrum Scale RAID: Administration* in Elastic Storage Server (ESS) documentation.

To run the system health check for verifying IBM Spectrum Scale installation, issue the following command:

```
mmhealth node show -N all
```

Manually installing IBM Spectrum Scale for object storage on Red Hat Enterprise Linux

IBM Spectrum Scale for object storage is typically installed by using the installation toolkit. If you do not want to use the installation toolkit, manually install IBM Spectrum Scale for object storage as follows.

For information about prerequisites, see [“Software requirements”](#) on page 220 and [“Installation prerequisites”](#) on page 352.

Before you begin manually installing IBM Spectrum Scale for object storage, complete the following prerequisite tasks.

1. Create protocol nodes for object service. For more information, see *Configuring CES protocol service IP addresses* in *IBM Spectrum Scale: Administration Guide*.
2. Add at least one CES IP address.

Manually install the object protocol and then enable object services as follows.

1. On all protocol nodes, install the `spectrum-scale-object` package and its associated dependencies by issuing the following command.

```
yum -y install spectrum-scale-object
```

The package is created in the `/usr/lpp/mmfs/5.1.5.x/object_rpms/rhel8` directory by expanding the IBM Spectrum Scale installation image. For more information about extracting an installation image, see [“Extracting the IBM Spectrum Scale software on Linux nodes” on page 355](#).

Note:

- The path `/usr/lpp/mmfs/5.1.5.x/object_rpms/rhel8` depends upon the release version.
- The Object protocol in IBM Spectrum Scale 5.1.2.0 and earlier 5.1.x releases requires OpenStack 16 repositories to be available on all protocol nodes to satisfy the necessary dependencies. For information on how to set up these repositories, see [“OpenStack repository configuration required by the object protocol” on page 319](#).

You might need to create the **yum** repository before you use this command. To create the **yum** repository, create an entry similar to the following entry in the `/etc/yum.repos.d` directory:

```
[spectrum_scale]
name=spectrum_scale
baseurl=file:///usr/lpp/mmfs/5.1.5.x/object_rpms/rhel8
enabled=1
gpgcheck=0
```

Note: The path `file:///usr/lpp/mmfs/5.1.5.x/object_rpms/rhel8`, in the preceding example depends upon the release version.

2. From one of the protocol nodes, configure the object protocol by using the **mmobj swift base** command.

```
mmobj swift base -g /ibm/fs1 -o object_fileset --cluster-hostname protocol-  
cluster.example.net \  
--local-keystone --pwd-file mmobjpwd
```

A sample output is as follows.

```
mmobj swift base: Validating execution environment.
mmobj swift base: Performing SELinux configuration.
mmobj swift base: Creating fileset /dev/fs1 object_fileset.
mmobj swift base: Configuring Keystone server in /gpfs/fs1/object/keystone.
mmobj swift base: Creating postgres database.
mmobj swift base: Validating Keystone environment.
mmobj swift base: Validating Swift values in Keystone.
mmobj swift base: Configuring Swift services.
mmobj swift base: Uploading configuration changes to the CCR.
mmobj swift base: Configuration complete.
```

The `mmobjpwd` value represents a file with the passwords that you can use in your configuration. For the password file specification format, see *mmobj command*. You can also pass configuration flags to `mmobj` to add support for features such as:

- S3 capability
- Unified file
- Object access
- Storage policies

The `mmobj` command page has the full list of available features that you can use.

After the initial installation and configuration is complete, the object protocol needs to be enabled across all the protocol nodes.

3. Complete the configuration and start object services on all protocol nodes by using the **mmces service enable** command.

```
# mmces service enable OBJ
```

4. List the protocols that are enabled in the IBM Spectrum Scale cluster by using the **mmces service list** command. List a verbose output of object services that are running on the local node by using the -v flag.

```
# mmces service list -v
```

A sample output is as follows.

```
Enabled services: OBJ SMB NFS
OBJ is running
OBJ:openstack-swift-object          is running
OBJ:openstack-swift-account         is running
OBJ:openstack-swift-container       is running
OBJ:openstack-swift-proxy           is running
OBJ:memcached                       is running
OBJ:openstack-swift-object-replicator is running
OBJ:openstack-swift-account-reaper  is running
OBJ:openstack-swift-account-replicator is running
OBJ:openstack-swift-container-replicator is running
OBJ:openstack-swift-object-sof      is running
OBJ:htpdp (keystone)               is running
SMB is running
NFS is running
```

5. You can enable new object storage features or manage existing object storage features including S3 capability, unified file and object access, and storage policies. For more information, see the *mmobj* command in the *IBM Spectrum Scale: Command and Programming Reference*.

Manually installing the performance monitoring tool

The performance monitoring tool is automatically installed by the installation toolkit. You can also install the performance monitoring tool manually.

- For information about the performance monitoring tool, see *Configuring the performance monitoring tool* in *IBM Spectrum Scale: Administration Guide*.
- For information about packages required before installation, see “Software requirements” on page 220.
- For information about the installation toolkit, see “Understanding the installation toolkit options” on page 392.

A single collector can easily support up to 150 sensor nodes. The collector can be any node on the system. All sensors report to this node. Select any node in the system to be the collector node. For information on configuring the sensors and collectors, see the *Configuring the performance monitoring tool* section in the *IBM Spectrum Scale: Problem Determination Guide*. You can manually install the tool by doing the following steps:

1. Download the installation images and install the performance monitoring packages.

For information on the default directories in which these packages are extracted, see “Location of extracted packages” on page 361.

Performance monitoring packages

```
gpfs.gss.pmsensors-version_release.os.target_arch.file_format
gpfs.gss.pmcollector-version_release.os.target_arch.file_format
```

Then, use your operating system’s native package management mechanism to install the packages.

For example:

- To install performance monitoring sensors on a Red Hat Enterprise Linux 7.x x86_64 node, use the following command:

```
rpm -ivh gpfs.gss.pmsensors-x.x.x-el7.x86_64.rpm
```


- To install performance monitoring sensors on a Red Hat Enterprise Linux 8.x x86_64 node, use the following command:

```
rpm -ivh gpfs.gss.pmsensors-x.x.x-x.el8.x86_64.rpm
```

- To install performance monitoring sensors on a SLES 15 x86_64 node, use the following command:

```
rpm -ivh gpfs.gss.pmsensors-x.x.x-x.sles12.x86_64.rpm
```

- To install performance monitoring sensors on an Ubuntu amd64 node, use the following command:

```
dpkg -i gpfs.gss.pmsensors_x.x.x-x.U*amd64.deb
```

- To install a performance monitoring collector on a Red Hat Enterprise Linux 7.x PPC64LE node, use the following command:

```
rpm -ivh gpfs.gss.pmcollector-x.x.x-x.el7.ppc64le.rpm
```

2. To configure performance monitoring after the packages are installed on all the selected nodes, use the following commands:

```
mmperfmon config generate --collectors collectorNode1[,collectorNode2,...]
mmchnode --perfmon -N sensorNode1[,sensorNode2...]
```

3. To enable performance monitoring on a protocol node for NFS, SMB, or Object, follow the given steps.

- For NFS:

- a. Install the pm-ganesha package as follows:

- RHEL or SLES: `rpm -ivh gpfs.pm-ganesha_version-release.os.target_arch.rpm`
- Ubuntu: `dpkg -i gpfs.pm-ganesha_version-release.deb`

- b. Ensure that the `/opt/IBM/zimon/defaults/ZIMonSensors_nfs.cfg` sensor is created as shown:

```
sensors={
    name = "NFSIO"
    period = 10
    type = "Generic"
    restrict = "cesNodes"
}
```

- c. Run the following command to add the sensor:

```
mmperfmon config add --sensors /opt/IBM/zimon/defaults/ZIMonSensors_nfs.cfg
```

- d. Run the following code to ensure that the newly added sensor is made visible to the system:

```
mmhealth node show nfs --refresh -N cesNodes
```

- For SMB:

- a. Run the following command to add the sensor:

```
mmperfmon config add --sensors /opt/IBM/zimon/defaults/ZIMonSensors_smb.cfg
```

Note: No additional sensor package is needed for the SMB sensors.

- b. Run the following code to ensure that the newly added sensor is made visible to the system:

```
mmhealth node show smb --refresh -N cesNodes
```

- For Object:

- a. Install the pmswift package as follows:

- RHEL: `rpm -ivh pmswift-version-release.noarch.rpm`

- Ubuntu: `dpkg -i pmswift_version-release.deb`

Where *version* is equal to or greater than 4.2 and *release* is equal to or greater than 0.

The installation of the pmswift RPM also copies SWIFT related sensors configuration files, namely, `SwiftAccount.cfg`, `SwiftContainer.cfg`, `SwiftObject.cfg`, and `SwiftProxy.cfg` to the performance monitoring tool's installation directory, `/opt/IBM/zimon/`. The pmswift package converts the operational metrics for Object into a form that is usable by the performance monitoring tool.

- b. Edit the Object configuration files for all Object servers that reside in the cluster configuration repository (CCR), using the following command:

```
/usr/local/pmswift/bin/pmswift-config-swift set
```

CCR then propagates the modified configuration files to `/etc/swift/` directory on all the protocol nodes within the cluster. The modified configuration files are:

- `account - *.conf`
- `container - *.conf`
- `object - *.conf`
- `proxy - *.conf`

- c. Use the `/usr/local/pmswift/bin/pmswift-config-zimon set` command to edit the sensors configuration information stored in the CCR. This adds the SWIFT related following sensors entries:

```
{
    # SwiftAccount operational metrics
    name = "SwiftAccount"
    period = 1
    type = "generic"
},
{
    # SwiftContainer operational metrics
    name = "SwiftContainer"
    period = 1
    type = "generic"
},
{
    # SwiftObject operational metrics
    name = "SwiftObject"
    period = 1
    type = "generic"
},
{
    # SwiftProxy operational metrics
    name = "SwiftProxy"
    period = 1
    type = "generic"
},
}
```

These entries are then automatically propagated to the `ZIMonSensors.cfg` file in `/opt/IBM/zimon` on all the nodes in the cluster.

- d. Start the **pmswiftd.service** by using the following command:

```
systemctl start pmswiftd.service
```

- e. Start or restart the **pmsensors.service** by using the following command:

```
systemctl start|restart pmsensors.service
```

For more information on how to manually upgrade pmswift, see the [“Manually upgrading pmswift”](#) on page 527 topic.

4. If the protocol sensors are enabled on a GPFS-only node, you see an error message stating that the sensors are unavailable. However, the other sensors continue to run.
5. Start the sensors on each node using the **systemctl start pmsensors.service** command.

6. On the collector nodes, start the collector, using the **systemctl start pmcollector.service** command.
7. To ensure that sensors and collectors are restarted after the node reboots, you can enable them using the following commands:

Sensors

To enable sensors, use the **systemctl enable pmsensors.service** command.

To disable sensors, use the **systemctl disable pmsensors.service** command.

Collector

To enable the collector, use the **systemctl enable pmcollector.service** command.

To disable the collector, use the **systemctl disable pmcollector.service** command.

The collector node starts gathering all the requested metrics.

Important: Once the packages are installed, `pmsensor` and `pmcollector` services are activated automatically.

Note: Although you can enable sensors on every node in a system, with the increase in number of nodes, the metric collection work for the collector also increases. It is recommended to ensure that collection of metrics does not increase above 1000000 metrics per second.

By default, the installation toolkit enables sensors on each protocol node (CES node) but not on the other GPFS nodes (non-CES nodes).

Metrics can be retrieved from any node in the system by using the **mmperfmon query** command.

For more information, see *mmperfmon command* in *IBM Spectrum Scale: Command and Programming Reference*.

For more information about the performance monitoring tool, see *Configuring the performance monitoring tool* in *IBM Spectrum Scale: Problem Determination Guide*.

Manually installing IBM Spectrum Scale management GUI

The management GUI provides an easy way for the users to configure, manage, and monitor the IBM Spectrum Scale system.

You can install the management GUI by using the following methods:

- Installing management GUI by using the installation toolkit. For more information, see [“Installing IBM Spectrum Scale management GUI by using the installation toolkit” on page 426](#).
- Manual installation of the management GUI. The following sections provide the details of how to manually install the management GUI.

Prerequisites

The prerequisites that are applicable for installing the IBM Spectrum Scale system through CLI are applicable for GUI installation as well. For more information, see [“Installation prerequisites” on page 352](#).

The IBM Spectrum Scale GUI package is also part of the installation package. You need to extract this package to start the installation. The performance tool packages enable the performance monitoring tool that is integrated into the GUI. The following packages are important for performance monitoring tools in GUI:

- The performance tool collector package. This package is placed only on the collector nodes. By default, every GUI node is also used as the collector node to receive performance details and display them in the GUI.
- The performance tool sensor package. This package is applicable for the sensor nodes, if not already installed.
- *iptables* is required for the installation of Linux operating systems. However, it might not be a prerequisite if the administrator configures the firewall for the specific GUI node. For more information, see *Firewall recommendations for IBM Spectrum Scale GUI* in *IBM Spectrum Scale: Administration Guide*.

Note: For GUI installations on RHEL9 you must install *nftables*. You can run the **dnf install nftables** command to install *nftables*.

Note: The GUI must be a homogeneous stack. That is, all packages must be of the same release. For example, do not mix the 5.1.2 GUI rpm with a 5.1.3 base rpm. However, GUI PTFs and efexes can usually be applied without having to install the corresponding PTF or efex of the base package. A GUI PTF and efex is helpful if you want to get rid of a GUI issue without changing anything on the base layer.

The following table lists the IBM Spectrum Scale GUI and performance tool package that are essential for different platforms.

<i>Table 36. GUI packages essential for each platform</i>	
GUI Platform	Package name
Red Hat Enterprise Linux (RHEL) 7.x and 8.x	gpfs.gui-5.1.5-x.noarch.rpm gpfs.java-5.1.5-x.x86_64.rpm gpfs.java-5.1.5-x.ppc64le.rpm gpfs.java-5.1.5-x.s390x.rpm
SUSE Linux Enterprise Server (SLES) 15	gpfs.gui-5.1.5-x.noarch.rpm gpfs.java-5.1.5-x.x86_64.rpm gpfs.java-5.1.5-x.ppc64le.rpm gpfs.java_5.1.5-x_s390x.rpm
Ubuntu 20	gpfs.gui_5.1.5-x_all.deb gpfs.java_5.1.5-x_amd64.deb gpfs.java_5.1.5-x_ppc64el.deb
Performance monitoring tool platform	Performance monitoring tool rpms
RHEL 8.x x86	gpfs.gss.pmcollector-5.1.5-x.el8.x86_64.rpm gpfs.gss.pmsensors-5.1.5-x.el8.x86_64.rpm
RHEL 7.x x86	gpfs.gss.pmcollector-5.1.5-x.el7.x86_64.rpm gpfs.gss.pmsensors-5.1.5-x.el7.x86_64.rpm
RHEL 7 s390x	gpfs.gss.pmsensors-5.1.5-x.el7.s390x.rpm gpfs.gss.pmcollector-5.1.5-x.el7.s390x.rpm
RHEL 8.x ppc64 LE	gpfs.gss.pmcollector-5.1.5-x.el8.ppc64le.rpm gpfs.gss.pmsensors-5.1.5-x.el8.ppc64le.rpm

Table 36. GUI packages essential for each platform (continued)	
GUI Platform	Package name
RHEL 7.x ppc64 LE	gpfs.gss.pmcollector-5.1.5-x.el7.ppc64le.rpm gpfs.gss.pmsensors-5.1.5-x.el7.ppc64le.rpm
SLES 15 x86	gpfs.gss.pmcollector-5.1.5-x.SLES15.x86_64.rpm gpfs.gss.pmsensors-5.1.5-x.SLES15.X86_64.rpm
SLES 15 s390x	gpfs.gss.pmsensors-5.1.5-x.SLES15.s390x.rpm gpfs.gss.pmcollector-5.1.5-x.SLES15.s390x.rpm
Ubuntu 20.04 LTS sensor and collector packages	gpfs.gss.pmsensors_5.1.5-x.U20.04_amd64.deb gpfs.gss.pmcollector_5.1.5-x.U20.04_amd64.deb

Ensure that the performance tool collector runs on the same node as the GUI.

Yum repository setup

You can use yum repository to manually install the GUI rpm files. It is the preferred way of GUI installation as yum checks the dependencies and automatically installs missing platform dependencies like the postgres module, which is required but not included in the package.

Installation steps

You can install the management GUI either by using the package manager (yum or zypper commands) or by installing the packages individually.

Installing management GUI by using package manager (yum or zypper commands)

It is recommended to use this method as the package manager checks the dependencies and automatically installs missing platform dependencies. Issue the following commands to install management GUI:

Red Hat Enterprise Linux

```
yum install gpfs.gss.pmsensors-5.1.5-x.elx.<arch>.rpm
yum install gpfs.gss.pmcollector-5.1.5-x.elx.<arch>.rpm
yum install gpfs.java-5.1.5-x.<arch>.rpm
yum install gpfs.gui-5.1.5-x.noarch.rpm
```

SLES

```
zypper install gpfs.gss.pmsensors-5.1.5-x.SLES15.<arch>.rpm
zypper install gpfs.gss.pmcollector-5.1.5-x.SLES15.<arch>.rpm
zypper install gpfs.java-5.1.5-x.<arch>.rpm
zypper install gpfs.gui-5.1.5-x.noarch.rpm
```

Installing management GUI by using RPM

Issue the following commands for both RHEL and SLES platforms:

```
rpm -ivh gpfs.java-5.1.5-x.<arch>.rpm
rpm -ivh gpfs.gss.pmsensors-5.1.5-x.elx.<arch>.rpm
rpm -ivh gpfs.gss.pmcollector-5.1.5-x.elx.<arch>.rpm
rpm -ivh gpfs.gui-5.1.5-x.noarch.rpm
```

Installing management GUI on Ubuntu by using dpkg and apt-get

Issue the following commands for Ubuntu platforms:

```
dpkg -i gpfs.java_5.1.5-x_<arch>.deb
dpkg -i gpfs.gss.pmsensors_5.1.5-x_<os>_<arch>.deb
dpkg -i gpfs.gss.pmcollector_5.1.5-x_<os>_<arch>.deb
apt-get install postgresql
dpkg -i gpfs.gui_5.1.5-x_all.deb
```

The sensor package must be installed on any additional node that you want to monitor. All sensors must point to the collector node.

Note: In IBM Spectrum Scale 5.1.4 you can disable the GUI service after you have installed it without restricting your access to the REST API service. You can configure `WEB_GUI_ENABLED = false` in the `gpfsGui.properties` file to access only the REST API service. The **WEB_GUI_ENABLED** parameter value is set to `true` by default.

Start the GUI

Start the GUI by issuing the **systemctl start gpfsGui** command.

Note: 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.

Enabling performance tools in management GUI

The performance tool consists of sensors that are installed on all nodes that need to be monitored. It also consists of one or more collectors that receive data from the sensors. The GUI expects that a collector runs on a GUI node. The GUI queries the collectors for performance and capacity data. The following steps use the automated approach to configure and maintain performance data collection by using the **mmperfmon** CLI command. Manually editing the `/opt/IBM/zimon/ZIMonSensors.cfg` file is not compatible with this configuration mode.

1. Install the necessary software packages. Install the collector software package, `gpfs.gss.pmcollector`, on all GUI nodes. Install the sensor software packages, `gpfs.gss.pmsensors`, on all nodes, which are supposed to send the performance data.
2. Initialize the performance collection. Use the **mmperfmon config generate --collectors [node list]** command to create an initial performance collection setup on the selected nodes. The GUI nodes must be configured as collector nodes. Depending on the installation type, this configuration might be already completed before. However, verify the existing configuration.
3. Enable nodes for performance collection. You can enable nodes to collect performance data by issuing the **mmchnode --perfmon -N [SENSOR_NODE_LIST]** command. **[SENSOR_NODE_LIST]** is a comma-separated list of sensor nodes' hostnames or IP addresses and you can also use a node class. Depending on the type of installation, nodes might be configured for performance collection.
4. Review peer configuration for the collectors. The **mmperfmon config update** command updates the multiple collectors with the necessary configuration. The collector configuration is stored in the `/opt/IBM/zimon/ZIMonCollector.cfg` file. This file defines the collector peer configuration and the aggregation rules. If you are using only a single collector, you can skip this step. The GUI must have access to all data from each GUI node. For more information, see *Configuring the collector* in *IBM Spectrum Scale: Administration Guide*.
5. Review aggregation configuration for the collectors. The collector configuration is stored in the `/opt/IBM/zimon/ZIMonCollector.cfg` file. The performance collection tool is configured

with predefined rules on how data is aggregated when it gets older. By default, four aggregation domains are created as shown:

- A raw domain that stores the metrics uncompressed.
- A first aggregation domain that aggregates data to 30-second averages.
- A second aggregation domain that stores data in 15-minute averages.
- A third aggregation domain that stores data in 6-hour averages.

You must not change the default **aggregation configuration** as the already collected historical metric information might get lost. You cannot manually edit the `/opt/IBM/zimon/ZIMonCollector.cfg` file in the automated configuration mode.

In addition to the aggregation that is done by the performance collector, the GUI might request aggregated data based on the zoom level of the chart. For more information, see *Configuring the collector* in *IBM Spectrum Scale: Administration Guide*.

6. Configure the sensors. Several GUI pages display performance data that is collected with the help of performance monitoring tools. If data is not collected, the GUI shows the error messages like "No Data Available" or "Objects not found" in the performance charts. Installation by using the *spectrumscale installation toolkit* manages the default performance monitoring installation and configuration. The GUI help that is available on the various pages shows performance metric information. The GUI context-sensitive help also lists the sensor names.

The **Services > Performance Monitoring** page provides option to configure the sensor configuration and provides hints for collection periods and restriction of sensors to specific nodes.

You can also use the **mmperfmon config show** command in the CLI to verify the sensor configuration. Use the **mmperfmon config update** command to adjust the sensor configuration to match your needs. For more information, see *Configuring the sensor* in *IBM Spectrum Scale: Administration Guide*.

The local file `/opt/IBM/zimon/ZIMonSensors.cfg` can be different on every node and the system might change this path whenever a configuration change occurs. Therefore, this file must not be edited manually when you are using the automated configuration mode. During distribution of the sensor configuration, the restrict clause is evaluated and the period for all sensors is set to 0 in the `/opt/IBM/zimon/ZIMonSensors.cfg` file. The setting is defined for those nodes that did not match the restrict clause. You can check the local file to confirm that a restrict clause worked as intended.

Configuring capacity-related sensors to run on a single-node

Several capacity-related sensors must run only on a single node as they collect data for a clustered file system. For example, *GPFSDiskCap*, *GPFSFilesetQuota*, *GPFSFileset* and *GPFSPool*.

It is possible to automatically restrict these sensors to a single node. For new installations, capacity-related sensors are automatically configured to a single node where the capacity collection occurs. An updated cluster, which was installed before ESS 5.3.7 (IBM Spectrum Scale 5.0.5), might not be configured to use this feature automatically and must be reconfigured. To update the configuration, you can use the **mmperfmon config update SensorName.restrict=@CLUSTER_PERF_SENSOR** command, where **SensorName** values include *GPFSFilesetQuota*, *GPFSFileset*, *GPFSPool*, and *GPFSDiskCap*.

To collect file system and disk level capacity data on a single node that is selected by the system, run the following command to update the sensor configuration.

```
mmperfmon config update GPFSDiskCap.restrict=@CLUSTER_PERF_SENSOR
```

If the selected node is in the DEGRADED state, then the CLUSTER_PERF_SENSOR is automatically reconfigured to another node that is in the HEALTHY state. The performance monitoring service is restarted on the previous and currently selected nodes. For more information, see *Automatic assignment of single node sensors* in *IBM Spectrum Scale: Problem Determination Guide*.

Note: If the *GPFSDiskCap* sensor is frequently restarted, it can negatively impact the system performance. The *GPFSDiskCap* sensor can cause a similar impact on the system performance as the **mmddf** command. Therefore, to avoid using the **@CLUSTER_PERF_SENSOR** for any sensor in the **restrict** field of a single node sensor until the node stabilizes in the HEALTHY state, it is advisable to use a dedicated healthy node. If you manually configure the **restrict** field of the capacity sensors then you must ensure that all the file systems on the specified node are mounted to record file system-related data, like capacity.

Use the **Services > Performance Monitoring** page to select the appropriate data collection periods for these sensors.

For the *GPFSDiskCap* sensor, the recommended period is 86400, which means once per day. As the *GPFSDiskCap.period* sensor runs **mmddf** command to get the capacity data, it is not recommended to use a value less than 10800 (every 3 hours). To show fileset capacity information, it is necessary to enable quota for all file systems where fileset capacity must be monitored. For more information, see the **-q** option in the **mmchfs** command and **mmcheckquota** command.

To update the sensor configuration for triggering an hourly collection of capacity-based fileset capacity information, run the **mmperfmon** command as shown in the following example,

```
mmperfmon config update GPFSFilesetQuota.restrict=@CLUSTER_PERF_SENSOR gui_node
GPFSFilesetQuota.period=3600
```

Checking GUI and performance tool status

Issue the **systemctl status gpfsgui** command to know the GUI status as shown in the following example.

```
systemctl status gpfsgui.service
gpfsgui.service - IBM_GPFS_GUI Administration GUI
Loaded: loaded (/usr/lib/systemd/system/gpfsgui.service; disabled)
Active: active (running) since Fri 2015-04-17 09:50:03 CEST; 2h 37min ago
Process: 28141 ExecStopPost=/usr/lpp/mmfs/gui/bin/cfgmantraclient unregister (code=exited, status=0/SUCCESS)
Process: 29120 ExecStartPre=/usr/lpp/mmfs/gui/bin/check4pgsql (code=exited, status=0/SUCCESS)
Main PID: 29148 (java)
Status: "GSS/GPFS GUI started"
CGroup: /system.slice/gpfsgui.service
<<<29148 /opt/ibm/wlp/java/jre/bin/java -XX:MaxPermSize=256m -Dcom.ibm.gpfs.platform=GPFS
-Dcom.ibm.gpfs.vendor=IBM -Djava.library.path=/opt/ibm/wlp/usr/servers/gpfsgui/lib/
-javaagent:/opt/ibm/wlp/bin/tools/ws-javaagent.jar -jar /opt/ibm/wlp/bin/tools/ws-server.jar
gpfsgui
--clean

Apr 17 09:50:03 server-21.localnet.com java[29148]: Available memory in the JVM: 484MB
Apr 17 09:50:03 server.localnet.com java[29148]: Max memory that the JVM will attempt to use:
512MB
Apr 17 09:50:03 server.localnet.com java[29148]: Number of processors available to JVM: 2
Apr 17 09:50:03 server.localnet.com java[29148]: Backend started.
Apr 17 09:50:03 server.localnet.com java[29148]: CLI started.
Apr 17 09:50:03 server.localnet.com java[29148]: Context initialized.
Apr 17 09:50:03 server.localnet.com systemd[1]: Started IBM_GPFS_GUI Administration GUI.
Apr 17 09:50:04 server.localnet.com java[29148]: [AUDIT ] CWWKZ0001I: Application /
started in 6.459 seconds.
Apr 17 09:50:04 server.localnet.com java[29148]: [AUDIT ] CWWKF0012I: The server
installed the following features: [jdbc-4.0, ssl-1.0, localConnector-1.0, appSecurity-2.0,
jsp-2.2, servlet-3.0, jndi-1.0, usr:FscUserRepo, distributedMap-1.0].
Apr 17 09:50:04 server-21.localnet.com java[29148]: [AUDIT ] CWWKF0011I: ==> When you see
the service was started anything should be OK !
```

Issue the **systemctl status pmcollector** and **systemctl status pmsensors** commands to know the status of the performance tool.

You can also check whether the performance tool backend can receive data by using the GUI. As another option, you can also use a command-line performance tool that is called **zc**, which is available in **/opt/IBM/zimon** folder. For example,

```
echo "get metrics mem_active, cpu_idle, gpfs_ns_read_ops last 10 bucket_size 1" | ./zc 127.0.0.1
Result example:
1: server-21.localnet.com|Memory|mem_active
2: server-22.localnet.com|Memory|mem_active
```



```

3: server-23.localnet.com|Memory|mem_active
4: server-21.localnet.com|CPU|cpu_idle
5: server-22.localnet.com|CPU|cpu_idle
6: server-23.localnet.com|CPU|cpu_idle
7: server-21.localnet.com|GPFSNode|gpfs_ns_read_ops
8: server-22.localnet.com|GPFSNode|gpfs_ns_read_ops
9: server-23.localnet.com|GPFSNode|gpfs_ns_read_ops
Row Timestamp mem_active mem_active mem_active cpu_idle cpu_idle cpu_idle gpfs_ns_read_ops
gpfs_ns_read_ops gpfs_ns_read_ops
1 2015-05-20 18:16:33 756424 686420 382672 99.000000 100.000000 95.980000 0 0 0
2 2015-05-20 18:16:34 756424 686420 382672 100.000000 100.000000 99.500000 0 0 0
3 2015-05-20 18:16:35 756424 686420 382672 100.000000 99.500000 100.000000 0 0 6
4 2015-05-20 18:16:36 756424 686420 382672 99.500000 100.000000 100.000000 0 0 0
5 2015-05-20 18:16:37 756424 686520 382672 100.000000 98.510000 100.000000 0 0 0
6 2015-05-20 18:16:38 774456 686448 384684 73.000000 100.000000 96.520000 0 0 0
7 2015-05-20 18:16:39 784092 686420 382888 86.360000 100.000000 52.760000 0 0 0
8 2015-05-20 18:16:40 786004 697712 382688 46.000000 52.760000 100.000000 0 0 0
9 2015-05-20 18:16:41 756632 686560 382688 57.580000 69.000000 100.000000 0 0 0
10 2015-05-20 18:16:42 756460 686436 382688 99.500000 100.000000 100.000000 0 0 0

```

Node classes used for the management GUI

The IBM Spectrum Scale management GUI automatically creates the following node classes during installation:

- **GUI_SERVERS:** Contains all nodes with a server license and all the GUI nodes
- **GUI_MGMT_SERVERS:** Contains all GUI nodes

Each node on which the GUI services are started is added to these node classes.

Nodes can also be removed from Node classes. For more information, see [“Removing nodes from management GUI-related node class” on page 503](#).

For more information about node classes, see *Specifying nodes as input to GPFS commands* in *IBM Spectrum Scale: Administration Guide*.

Related concepts

[“Uninstalling the IBM Spectrum Scale management GUI” on page 502](#)

Do the following to uninstall management GUI and remove the performance monitoring components that are installed for the GUI:

[“IBM Spectrum Scale GUI” on page 145](#)

You can configure and manage various features that are available with the IBM Spectrum Scale system by using the IBM Spectrum Scale management GUI.

Root privilege considerations for IBM Spectrum Scale management GUI

The IBM Spectrum Scale GUI WebSphere Java process runs as a user named "scalemgmt". This provides improved security because web applications running as a non-root user are less vulnerable to security threats. The *scalemgmt* user is set up as a system account with no login privileges.

The GUI and GUI user still need root privileges in the backend to perform the following tasks:

- Monitoring and managing the cluster by issuing the IBM Spectrum Scale CLI commands.
- Bind to the privileged ports such as 80 (http) and 443 (https) to work with the GUI.

The system automatically performs the required configuration. It does not require any configuration to be done by the user.

Enabling the scalemgmt user to monitor and manage the system through GUI

As root privileges are not available to the GUI user, the system enables the scalemgmt user to run the CLI commands through sudo. The GUI installation adds the file `/etc/sudoers.d/scalemgmt_sudoers`, which allows the scalemgmt user to run commands that match the wildcard `"/usr/lpp/mmfs/bin/mm"`.

During installation of the IBM Spectrum Scale management GUI, the following lines are appended to the `/etc/sudoers` file:

```
## Read drop-in files from /etc/sudoers.d
#includedir /etc/sudoers.d
```

Note: Do not alter these lines if you are editing the `/etc/sudoers` file. If these lines are removed or moved from the default location, the GUI cannot run the CLI commands as a non-root user.

Binding to the GUI privileged ports

A non-root process cannot bind privileged ports such as 80 and 443. So, the GUI process now runs on port 47443 and 47080 and uses *iptables* rules to forward port 443 to 47443 and 80 to 47080. The GUI is only available in the browser on port 443 through HTTPS. However, versions prior to 4.2.3 used port 80 to send events to the GUI, so to stay backwards compatible in clusters that still contain nodes with older versions, the GUI still accepts events on port 80.

To reach the GUI from outside, not only port 443 and 80 but also port 47443 and 47080 must be opened to the outside. This is because the port forwarding redirects port 443 to 47443 and 80 to 47080 before evaluating the rules of the *iptables* INPUT chain.

The *iptables* rules necessary for the port forwarding are automatically checked every time the GUI is started through the **systemctl start gpfsgui** command. The user does not have to configure anything manually.

Installing IBM Spectrum Scale on Linux nodes with the installation toolkit

You can install IBM Spectrum Scale by using the installation toolkit. You can also use the installation toolkit for tasks such as deploying protocols on existing clusters and adding nodes to an existing IBM Spectrum Scale cluster.

Overview of the installation toolkit

The installation toolkit automates the steps that are required to install IBM Spectrum Scale, deploy protocols, and upgrade to a later IBM Spectrum Scale release. For a release-wise list of features available in the installation toolkit, see [Table 37 on page 388](#).

When using the installation toolkit, you provide environmental information based on which the installation toolkit dynamically creates a cluster definition file. Thereafter, the installation toolkit installs, configures, and deploys the specified configuration.

The installation toolkit enables you to do the following tasks:

- Install and configure IBM Spectrum Scale.
- Add IBM Spectrum Scale nodes to an existing cluster.
- Deploy and configure SMB, NFS, Object, HDFS, and performance monitoring tools.
- Perform verification before installing, deploying, or upgrading. It includes checking whether passwordless SSH is set up correctly.
- Enable and configure call home and file audit logging functions.
- Upgrade IBM Spectrum Scale.

Installation and configuration are driven through commands.

From the self-extracting package, the installation toolkit is extracted to this directory, by default:

```
/usr/lpp/mmfs/package_code_version/ansible-toolkit
```

Using the installation toolkit is driven through the **spectrumscale** command in this directory, and this directory can optionally be added to the path.

The installation toolkit operation consists of four phases:

1. User input by using **spectrumscale** commands

- a. All user input is recorded into a cluster definition file in `/usr/lpp/mmfs/5.1.5.0/ansible-toolkit/ansible/vars`.
- b. Review the cluster definition file to make sure that it accurately reflects your cluster configuration.
- c. As you input your cluster configuration, you can have the installation toolkit act on parts of the cluster by not specifying nodes that might have incompatible operating systems, OS versions, or architectures.

2. A **spectrumscale install** phase

- a. Installation acts upon all nodes that are defined in the cluster definition file.
- b. GPFS and performance monitoring packages are installed.
- c. File audit logging and AFM to cloud object storage packages might be installed.
- d. GPFS portability layer is created.
- e. GPFS is started.
- f. A GPFS cluster is created.
- g. Licenses are applied.
- h. GUI nodes might be created and the GUI might be started upon these nodes.
- i. Performance monitoring, GPFS ephemeral ports, and cluster profile might be configured.
- j. NSDs are created.
- k. File systems are created.

3. A **spectrumscale deploy** phase

- a. Deployment acts upon all nodes that are defined into the cluster definition file.
- b. SMB, NFS, HDFS, and Object protocol packages are copied to all protocol nodes and installed.
- c. SMB, NFS, HDFS, and Object services might be started.
- d. File audit logging and message queue might be configured.
- e. Licenses are applied.
- f. GUI nodes might be created and the GUI might be started upon these nodes.
- g. Performance monitoring, call home, file audit logging, GPFS ephemeral ports, and cluster profile might be configured.

4. A **spectrumscale upgrade** phase:

Note: The upgrade phase does not allow new function to be enabled. In the upgrade phase, the required packages are upgraded, but adding functions must be done either before or after the upgrade.

- a. Upgrade acts upon all nodes input into the cluster definition file. However, you can exclude a subset of nodes from the upgrade configuration.
- b. All installed or deployed components are upgraded. During the upgrade phase, any missing packages might be installed and other packages that are already installed are upgraded.
- c. Upgrade can be done in the following ways:
 - Online upgrade (one node at a time)
Online upgrades are sequential with multiple passes. For more information, see [“Upgrade process flow” on page 545](#).
 - Offline upgrade
Offline upgrades can be done in parallel saving a lot of time in the upgrade window.
 - Upgrade while excluding a subset of nodes

- d. Allows for prompting to be enabled on a node to pause to allow for application migration from the node before proceeding with upgrade.

For more information, see [“Upgrading IBM Spectrum Scale components with the installation toolkit” on page 543](#) and [“Upgrade process flow” on page 545](#).

For information about command options available with the **spectrumscale** command, see the spectrumscale command description in the *IBM Spectrum Scale: Command and Programming Reference*.

The following table lists the features available in the installation toolkit in the reverse chronological order of releases.

Table 37. Installation toolkit: List of features	
Release	Features
5.1.5.x	<ul style="list-style-type: none">• Ansible collection support in the toolkit.• Precheck problem determination enhancement.• Config populate enhancement.
5.1.4.x	<ul style="list-style-type: none">• Support for Ubuntu 22.04 on x86_64.• Support for Red Hat Enterprise Linux 8.6 on x86_64, PPC64LE, and s390x.• Modified the command to enable workload prompt to allow administrators to stop and migrate workloads before a node is shut down for upgrade. For more information, see “Upgrading IBM Spectrum Scale components with the installation toolkit” on page 543.
5.1.3.x	<ul style="list-style-type: none">• Support for parallel offline upgrade of all nodes in the cluster.• Support for Ansible 2.10.x.
5.1.2.x	<ul style="list-style-type: none">• Support for Red Hat Enterprise Linux 8.5 on x86_64, PPC64LE, and s390x.• Support for user-defined profiles of GPFS configuration parameters.• Support for populating cluster state information from mixed CPU architecture nodes. The information is populated from nodes that have the same CPU architecture as the installer node.• Several optimizations in the upgrade path resulting in faster upgrades than in earlier releases.

Table 37. Installation toolkit: List of features (continued)

Release	Features
5.1.1.x	<ul style="list-style-type: none"> • Migration to the Ansible® automation platform: <ul style="list-style-type: none"> – Enables scaling up to more number of nodes – Avoids issues that arise due to using an agent-based tooling infrastructure such as Chef – Enables parity with widely-adopted, modern tooling infrastructure • Support for Red Hat Enterprise Linux 8.4 on x86_64, PPC64LE, and s390x • Support for Ubuntu 20.04 on PPC64LE • Support for multiple recovery groups in IBM Spectrum Scale Erasure Code Edition • Simplification of file system creation: <p>The file system is created during the installation phase rather than the deployment phase.</p> • Support for IPv6 addresses • Support for the CES interface mode • IBM Spectrum Scale deployment playbooks are now open sourced on GitHub. Users can access the playbooks from the external GitHub repository and implement in their environment on their own. For more information, see https://github.com/IBM/ibm-spectrum-scale-install-infra. <p>Note: The installation toolkit (./spectrumscale command) is only available as part of the IBM Spectrum Scale installation packages that can be downloaded from IBM FixCentral.</p> <ul style="list-style-type: none"> • Added an option to enable workload prompt to allow administrators to stop and migrate workloads before a node is shut down for upgrade. • Discontinued the following functions: <ul style="list-style-type: none"> – NTP configuration <p>Time synchronization configuration across all nodes in a cluster is recommended. Do it manually by using the available method.</p> – File and object authentication configuration <p>File and object authentication configuration must be done by using the mmuserauth command.</p> – NSD balance <p>Balance the NSD preferred node between the primary and secondary nodes by using the ./spectrumscale nsd servers command.</p>

Table 37. Installation toolkit: List of features (continued)

Release	Features
5.1.0.x	<ul style="list-style-type: none"> • Support for Ubuntu 20.04 on x86_64 • Support for Red Hat Enterprise Linux 8.3 on x86_64, PPC64LE, and s390x • Support for Red Hat Enterprise Linux 7.9 on x86_64, PPC64LE, and s390x • Support for NFS and SMB protocols, and CES on SLES 15 on x86_64 • Support for installing and upgrading AFM to cloud object storage (gpfs.afm.cos) package
5.0.5.x	<ul style="list-style-type: none"> • Support for Red Hat Enterprise Linux 8.2 on x86_64, PPC64LE, and s390x • Support for Red Hat Enterprise Linux 7.8 on x86_64, PPC64, PPC64LE, and s390x • Support for packages and repository metadata signed with a GPG (GNU Privacy Guard) key • Enhanced handling of host entries in the /etc/hosts file. Support for both FQDN and short name
5.0.4.x	<ul style="list-style-type: none"> • Support for Hadoop Distributed File System (HDFS) protocol • Support for ESS 3000 environments. • Support for Red Hat Enterprise Linux 8.0 and 8.1 on x86_64, PPC64LE, and s390x • Support for Red Hat Enterprise Linux 7.7 on x86_64, PPC64, PPC64LE, and s390x • Support for SLES 15 SP1 on x86_64 and s390x • Note: NFS, SMB, and object are not supported on SLES 15 SP1. • Improvements in online and offline upgrade paths • Removed the installation GUI • Support for IBM Spectrum Scale Developer Edition

Table 37. Installation toolkit: List of features (continued)

Release	Features
5.0.3.x	<ul style="list-style-type: none"> • Support for SLES 15 on x86_64 and s390x <p>Note: NFS, SMB, and object are not supported on SLES 15.</p> • Upgrade related enhancements: <ul style="list-style-type: none"> – Upgrade flow changes to minimize I/O disruptions – Enhanced upgrade pre-checks to determine the packages that must be upgraded. <p>Compare the versions of the installed packages with the versions in the repository of the packages you want to upgrade to. In a mixed operating system cluster, the comparison is done with the package repository applicable for the operating system running on the respective nodes.</p> <ul style="list-style-type: none"> – Mixed OS support for upgrade – Enhanced upgrade post-checks to ensure that all packages have been upgraded successfully – Enhanced dependency checks to ensure dependencies are met for each required package • IBM Spectrum Scale Erasure Code Edition <ul style="list-style-type: none"> – Ability to define a new setup type ece – Ability to designate a scale-out node – Ability to define recovery group, vdisk set, and file system – Support for installation and deployment of IBM Spectrum Scale Erasure Code Edition – Support for config populate function in an IBM Spectrum Scale Erasure Code Edition environment – Offline upgrade support for IBM Spectrum Scale Erasure Code Edition
5.0.2.x	<ul style="list-style-type: none"> • Support for IBM Z (RHEL 7.x, SLES12.x, Ubuntu 16.04 and Ubuntu 18.04 on s390x) • Support for RHEL 7.6 on x86_64, PPC64, PPC64LE, and s390x • Support for offline upgrade of nodes or components while they are stopped or down • Support for excluding nodes from an upgrade run • Support for rerunning an upgrade procedure after a failure • Support for watch folder • Configuration of message queue for file audit logging and watch folder • Enhancements in CES shared root creation and detection in config populate • Upgraded bundled Chef package

Table 37. Installation toolkit: List of features (continued)

Release	Features
5.0.1.x	<ul style="list-style-type: none"> • Support for Ubuntu 18.04 on x86_64 • Support for RHEL 7.5 on x86_64, PPC64, and PPC64LE • Support for Ubuntu 16.04.4 on x86_64 • Config populate support for call home and file audit logging • Performance monitoring configuration-related changes
5.0.0.x	<ul style="list-style-type: none"> • Extended operating system support <ul style="list-style-type: none"> – Ubuntu 16.04.0, 16.04.1, 16.04.2, 16.04.3 on x86_64 – RHEL 7.4 on x86_64, PPC64, and PPC64LE – SLES 12 SP3 on x86_64 • Improved deployment integration with Elastic Storage Server: The installation toolkit includes the capability to detect ESS nodes (EMS and I/O) and it ensures validation of permitted operations when you are adding protocol or client nodes to a cluster containing ESS. • File audit logging installation and configuration • Call home configuration • Cumulative object upgrade support • Enhanced network connectivity pre-checks including passwordless SSH validation from the admin node • Updated file system default block size for more likely best performance defaults

Understanding the installation toolkit options

Use the following information to understand how to work with the installation toolkit, and the options that are available with it.

Installation

When you use the installation toolkit to install IBM Spectrum Scale, the procedure comprises two stages:

1. Using a series of **spectrumscale** commands to add node and NSD specifications and configuration properties to the cluster definition file.
2. Using the **spectrumscale install** command to install IBM Spectrum Scale on the nodes specified in the cluster definition file and to apply the configuration options to the cluster.

Deployment

When you use the installation toolkit to deploy protocols, the procedure comprises two similar stages:

1. Using a series of **spectrumscale** commands to specify environment details, to identify which protocols are to be enabled, and to define protocol-specific properties in the cluster definition file.
2. Using the **spectrumscale deploy** command to deploy protocols as specified in the cluster definition file.

If you already have an existing IBM Spectrum Scale cluster, with IBM Spectrum Scale started and at least one file system for the CES shared file system, you can just define protocol nodes in the cluster definition and then deploy protocols.

Upgrade

When you use the installation toolkit to upgrade, the procedure comprises two similar stages:

1. Using a series of **spectrumscale** commands to specify environment details, or using **./spectrumscale config populate** to populate cluster wide properties in the cluster definition file in preparation for upgrade.
2. Using the **spectrumscale upgrade** command to upgrade all nodes and features as specified in the cluster definition file.

See “Using the installation toolkit to perform installation tasks: Explanations and examples” on page 407 for information on how to use the command options listed in Table 38 on page 393 to perform the installation and deployment. After you review the examples, you can tailor them according to your needs and then proceed to implement them.

<i>Table 38. spectrumscale command options for installing IBM Spectrum Scale and deploying protocols</i>	
spectrumscale command option	Purpose
node add	Adds node specifications to the cluster definition file
node delete	Removes node specifications from the cluster definition file
nsd add	Adds NSD specifications to the cluster definition file
nsd clear	Clears all NSDs
nsd delete	Removes a single NSD
nsd list	Lists all NSDs currently in the configuration
nsd modify	Modifies an NSD
filesystem list	Lists all file systems that currently have NSDs assigned to them
filesystem modify	Changes the block size and mount point of a file system
config gpfs	Adds IBM Spectrum Scale specific properties to the cluster definition file
install	Installs IBM Spectrum Scale on the configured nodes, creates a cluster, and creates NSDs.
config protocols	Provides details about the IBM Spectrum Scale environment to be used during protocol deployment
config perfmon	Configures performance monitoring settings
config object	Defines object-specific properties to be applied during deployment
config populate	Populates cluster definition file with current cluster configuration
deploy	Creates file systems and deploys protocols on your configured nodes
callhome	Configures call home settings
fileauditlogging	Configures file audit logging settings
upgrade	Upgrades the various components of the installation.

The list of **spectrumscale** command options listed is not exhaustive. For all command options available with the **spectrumscale** command and for more information about these command options, see the spectrumscale command description in the *IBM Spectrum Scale: Command and Programming Reference*.

Limitations of the installation toolkit

Before using the installation toolkit to install IBM Spectrum Scale and deploy protocols, review the following limitations and workarounds, if any.

Table 39. Installation toolkit limitations		
Function	Description	Workaround, if any
CES groups	The installation toolkit does not support the configuration of CES groups. This causes protocol deployments to fail in multi-network environments. Having CES groups might cause issues with upgrades as well because the installation toolkit is not aware of CES groups during the upgrade.	
Clusters larger than 64 nodes	The installation toolkit does not restrict the number of nodes in a cluster in which the toolkit can be used. However, it is designed as a single node server so as the number of nodes increases, the bandwidth to the installer node decreases, and the latency goes up. This implementation might cause issues in clusters with more than 64 nodes. Note: If you want to use the installation toolkit in a cluster larger than 64 nodes, contact scale@us.ibm.com .	
Clusters without passwordless SSH between all nodes	If clusters are set up in an AdminCentral=True configuration, which is a widely used configuration, the installation toolkit and protocols might not function correctly.	Set up passwordless SSH between all nodes in the cluster and to the nodes themselves using FQDN, IP address, and host name.
Compression	The installation toolkit does not configure file system compression.	After installation, configure the compression function manually. For more information, see <i>File compression</i> in <i>IBM Spectrum Scale: Administration Guide</i> .

Table 39. Installation toolkit limitations (continued)

Function	Description	Workaround, if any
Concurrent upgrade	<p>The installation toolkit does not support concurrent upgrade. You must plan for an outage depending on your setup. This brief outage prevents mixed code versions from running at the same time and it is also needed in a manual upgrade.</p> <p>Although the upgrade is non-concurrent, data is typically still accessible during the upgrade window. Access might be lost and need to be reestablished multiple times due to how the upgrade procedure is run among nodes.</p>	
Configuration change	<p>The installation toolkit does not support changing the configuration of entities that exist such as file systems and NSDs. For example, changing the block size or the default replication factor for data blocks for an existing file system is not supported. However, the installation toolkit can be used to add nodes, NSDs, or file systems to an existing cluster.</p> <p>The installation does not support authentication reconfiguration. It does not use the authentication section of the cluster definition file during upgrade.</p>	<ul style="list-style-type: none"> • If you want to change the configuration of existing file systems or NSDs, use the IBM Spectrum Scale commands after installation or deployment. • If you want to change the authentication method, see <i>Modifying authentication method</i> in <i>IBM Spectrum Scale: Administration Guide</i>.
Customer designation of sensor or collector nodes	The installation toolkit does not support customer designation of sensor or collector nodes. The installation toolkit automatically sets up sensor or collectors without allowing the user to choose which nodes have these functions.	<ol style="list-style-type: none"> 1. Use the -x flag to reconfigure performance monitoring, if needed. 2. Follow up with manual configuration of performance monitoring.
Disabling or uninstalling protocols and uninstalling GPFS	The installation toolkit does not support disabling or uninstalling protocols and uninstalling GPFS on an existing GPFS cluster.	<p>Use the manual procedures.</p> <ul style="list-style-type: none"> • For information about removing exports, see <i>Managing protocol data exports</i> in <i>IBM Spectrum Scale: Administration Guide</i>. • For information about disabling protocol services, see <i>Managing protocol services</i> in <i>IBM Spectrum Scale: Administration Guide</i>. • For information about uninstalling GPFS, see Chapter 15, “Steps to permanently uninstall IBM Spectrum Scale,” on page 495.

<i>Table 39. Installation toolkit limitations (continued)</i>		
Function	Description	Workaround, if any
ESS and IBM Spectrum Scale Erasure Code Edition mixed environment	The installation toolkit does not support an environment in which ESS and IBM Spectrum Scale Erasure Code Edition coexist.	Use the manual procedure to install, deploy, or upgrade IBM Spectrum Scale Erasure Code Edition in a cluster that contains ESS.
Encryption	The installation toolkit does not support encrypted file systems. Therefore, installation and deployment by using the installation toolkit do not work if the CES shared root file system or any other file system that the installation toolkit works with is encrypted.	
EPEL repositories	The installation toolkit does not support the configuration of protocols when EPEL repositories are configured.	Remove or disable EPEL repositories before you use the installation toolkit for installation, deployment, or upgrade.
File system DMAPI flag set to Yes (-z) installation and deployment	<p>The file system DMAPI flag is used in IBM Spectrum Scale for IBM Spectrum Protect for Space Management and policy management, attaching with an IBM Spectrum Protect server, and with IBM Spectrum Archive.</p> <p>The installation toolkit does not provide an option to add a DMAPI flag to a new or existing file system. If you need add a DMAPI flag to a file system, use the installation toolkit to create the file system and later set the DMAPI flag manually.</p>	For information on manual procedures for installation and deployment, see “Manually installing the IBM Spectrum Scale software packages on Linux nodes” on page 359.

Table 39. Installation toolkit limitations (continued)

Function	Description	Workaround, if any
File system DMAPI flag set to Yes (-z) upgrade	<p>An upgrade by using the installation toolkit is affected by the presence of the DMAPI flag in a few ways:</p> <ol style="list-style-type: none"> 1. If the CES shared root shares a file system that has the DMAPI flag set: <ul style="list-style-type: none"> • In this situation, the installation toolkit cannot unmount or mount the file system on each node it attempts to upgrade unless the user has preemptively removed the DMAPI flag and stopped all DMAPI services. For more information, see “Upgrading IBM Spectrum Scale on IBM Spectrum Archive Enterprise Edition (EE) nodes by using the installation toolkit” on page 562. • This scenario is different from scenario 2 because the CES shared root file system is essential for maintaining healthy CES status of the cluster. Without the CES shared root mounted, the CES component remains unhealthy and prevents upgrade from continuing. 2. If a non-CES shared root file system has the DMAPI flag set: <ul style="list-style-type: none"> • In this scenario, the installation toolkit cannot unmount the file system during upgrade of each node unless the user stopped all DMAPI-related services and unmounted the file system. • Thereafter, the installation toolkit cannot remount the file system after upgrade of each node. The file system is left unmounted and the user needs to bring it up manually on each node individually after the GPFS portion of the upgrade finished. Otherwise, the user can wait until the upgrade finishes and start the file system on all nodes at once. 	<ul style="list-style-type: none"> • For information on using the installation toolkit for upgrade in clusters where the DMAPI flag is set to Yes for any of the file systems, see “Upgrading IBM Spectrum Scale on IBM Spectrum Archive Enterprise Edition (EE) nodes by using the installation toolkit” on page 562. • In the case of a failed upgrade that needs to be restarted, if some of the nodes have been restarted, then steps need to be taken to get all of the file systems mounted again and the cluster healthy before another attempt can be made. This might require stopping DMAPI services again if they auto-started after a node was upgraded.

Table 39. Installation toolkit limitations (continued)

Function	Description	Workaround, if any
FPO configuration for disks	The installation toolkit does not support the extra stanza file flags required for FPO setup.	Do one of the following: <ol style="list-style-type: none"> 1. Create NSDs using the installation toolkit. 2. Manually edit the NSD stanza file afterward. The installation toolkit places the NSD stanza file in the <code>/usr/lpp/mmfs</code> directory. 3. Use mmchnsd to do the changes. OR <ol style="list-style-type: none"> 1. Create the cluster by using the installation toolkit. 2. Deploy protocols on the protocol nodes by using the installation toolkit. 3. Manually create the NSDs for the FPO setup.
Federal Information Processing Standard (FIPS) enabled	The installation toolkit does not support environments in which FIPS is enabled.	
GPG (GNU Privacy Guard) signed package support	The installation toolkit support for packages that are signed with the GPG key has the following limitation: <ul style="list-style-type: none"> • Repository metadata signing is not supported on Ubuntu. 	
Host-based SSH authentication	The installation toolkit does not support host-based SSH authentication. It supports only key-based SSH authentication.	Either set up key-based SSH authentication temporarily for use with the toolkit, or follow the manual steps in “Manually installing the IBM Spectrum Scale software packages on Linux nodes” on page 359.
Kafka packages	To make the IBM Spectrum Scale cluster ready for file audit logging and watch folder functions, the installation toolkit automatically installs the <code>gpfs.librdkafka</code> package for supported operating systems and hardware architectures. This might lead to errors if the prerequisites for <code>gpfs.librdkafka</code> are not installed.	Ensure that the prerequisite packages for <code>gpfs.librdkafka</code> are installed. For more information, see “Requirements, limitations, and support for file audit logging” on page 489.

Table 39. Installation toolkit limitations (continued)		
Function	Description	Workaround, if any
Local host	<ul style="list-style-type: none"> The installation toolkit does not support specifying the IP address that the local host resolves to, such as 127.0.0.1, as the installer node. The installation toolkit does not support adding the local host or the IP address that the local host resolves to, such as 127.0.0.1, as a node in the cluster configuration. 	
Multiple CES networks	The installation toolkit does not support deployments with multiple CES networks. Attempting deployment in this configuration has a high probability of failure. This is because when the CES address pool has multiple subnets, the command for adding CES address assigns an IP to a CES node that cannot handle that subnet of address, which causes the deployment failure.	
Multiple clusters	The installation toolkit does not support multiple clusters being defined in the cluster definition.	
Multi-region object deployment	For a multi-region object deployment, the installation toolkit only sets up the region number not the replication. For information about setting up multi-region object deployment, see “Enabling multi-region object deployment initially” on page 448.	
NFS or SMB exports configuration	The installation toolkit does not configure any exports on SMB or NFS.	Use the manual procedure. For information about configuring Cluster Export Services and creating exports, see <i>Configuring Cluster Export Services</i> and <i>Managing protocol data exports in IBM Spectrum Scale: Administration Guide</i> .
Node function addition during upgrade	The installation toolkit does not support designating node functionality during upgrade.	To add a function to the cluster or a node, designate this new function using the installation toolkit and proceed with an installation or a deployment. Perform this action either before or after an upgrade.
Non-English languages in client programs such as PuTTY	The installation toolkit does not support setting the language in client programs such as PuTTY to any language other than English.	Set the language of your client program to English.

Table 39. Installation toolkit limitations (continued)

Function	Description	Workaround, if any
NSD SAN attachment during initial installation	The installation toolkit cannot be used for NSD SAN attachment during initial installation because when adding NSDs using the installation toolkit, a primary and an optional comma-separated list of secondary NSD servers must be designated.	<ol style="list-style-type: none"> 1. Create NSDs using the installation toolkit. 2. Manually edit the NSD stanza file afterward. The NSD stanza files created by the installation toolkit are named in the <code>StanzaFile.xx</code> format and they are located in the <code>/usr/lpp/mmfs/</code> directory. 3. Use mmchnsd to do the changes.
Object protocol with IPv6 configured	The installation toolkit does not support object protocol in a cluster in which IPv6 is configured.	
Online upgrade of a 2-node cluster without tie-breaker disks	Online upgrade of a 2-node cluster that does not have tie-breaker disks configured is not supported. To do an online upgrade of a 2-node cluster by using the installation toolkit, tie-breaker disks must be configured in the cluster. A 1-node cluster can be upgraded only offline.	Do an offline upgrade. For more information, see “Performing offline upgrade or excluding nodes from upgrade by using installation toolkit” on page 554.
Package managers other than yum, zypper, or apt-get	The installation toolkit requires the use of yum (RHEL), zypper (SLES), and apt-get (Ubuntu) package managers to function.	
PPC and x86_64 or PPC and s390x or x86_64 and s390x mix	The installation toolkit does not support mixed CPU architecture configurations.	Use the installation toolkit on a subset of nodes that are supported and then manually install or deploy on the remaining nodes. For upgrading a mixed CPU architecture cluster, you can use the installation toolkit in two hops. For more information, see “Upgrading mixed CPU architecture cluster” on page 559.
Quorum or manager configuration after cluster installation	The installation toolkit allows a user to add <code>-m</code> (to specify a manager node) and <code>-q</code> (to specify a quorum node) flags to various nodes as they are added. If the proposed configuration does not match the existing configuration, the installation toolkit does nothing to change it.	Manually change node roles by using the mmchnode command.

Table 39. Installation toolkit limitations (continued)		
Function	Description	Workaround, if any
Remote mounted file systems	<ul style="list-style-type: none"> Object protocol deployment using the installation toolkit fails in case of remotely mounted file systems. This occurs because the object component must both list and create filesets, which is not allowed on a remotely mounted file system. NFS and SMB deployments on remote mounted file systems work using the installation toolkit. The setup of a CES shared root file system when the file system is remotely mounted works using the installation toolkit. 	
Repository proxy	The installation toolkit does not support proxy setups when working with repositories. <code>yum repolist</code> must not have any failed repos and it must be clean.	Ensure that there are no stale or failed repositories and that only the base OS repositories are enabled during any installation toolkit activities such as installation, deployment, or upgrade.
RPMs that have dependencies upon GPFS RPMs and GPFS settings	In an environment where some RPMs have dependencies on base GPFS RPMs or GPFS settings, the installation toolkit cannot be used for installation or upgrade.	
Running mmchconfig release=LATEST to complete an upgrade	The installation toolkit does not run mmchconfig release=LATEST after an upgrade. This is to give users time to verify an upgrade success and decide if the code level upgrade should be finalized.	Use mmchconfig release=LATEST after an upgrade using the installation toolkit to finalize the upgrade across the cluster.
Separate admin and daemon network	The installation toolkit does not support separate admin and daemon network.	
Sudo user	The installation toolkit does not function correctly unless run as root. Running as sudo or as another user does not work.	
Support for AIX, Debian, PowerKVM, Windows	The installation toolkit does not support AIX, Debian, PowerKVM, Windows operating systems. If these operating systems are installed on any cluster nodes, do not add these nodes to the installation toolkit.	Use the installation toolkit on a subset of nodes that are supported and then manually perform installation, deployment, or upgrade on the remaining nodes. For information about manual upgrade, see Chapter 16, “Upgrading,” on page 505.
Tie-Breaker NSD configuration	The installation toolkit does not configure tie-breaker disks.	Manually set the tie-breaker configuration as required using mmchconfig after completing installation using the toolkit.

Table 39. Installation toolkit limitations (continued)		
Function	Description	Workaround, if any
Transparent cloud tiering	The installation toolkit does not install, configure, or upgrade Transparent cloud tiering.	Use the manual procedures. For more information, see Chapter 7, “Installing Cloud services on IBM Spectrum Scale nodes,” on page 475.
Unique NSD device configuration	The installation toolkit relies upon a user having already configured and run the nsddevices sample script provided within a GPFS installation. The mmcrnsd and mmchnsd commands require running of the nsddevices script beforehand. Therefore, the installation toolkit will fail if this has not been done by the user.	
Upgrade while skipping over versions	The installation toolkit does not support skipping over major or minor versions of IBM Spectrum Scale releases when doing an online upgrade. For example, if an IBM Spectrum Scale cluster is at version 4.1.1.x, you cannot use the installation toolkit for an online upgrade directly to version 5.1.x.	Do an offline upgrade from version 4.1.1.x to 5.1.x. For more information, see “Performing offline upgrade or excluding nodes from upgrade by using installation toolkit” on page 554.

Related concepts

[“Limitations of config populate option of the installation toolkit ”](#) on page 427

Mixed operating system support with the installation toolkit

You can use the installation toolkit to install GPFS and deploy protocols in a cluster that contains nodes that are running on different operating systems. Several validations are done when you use the **spectrumscale** command in a mixed operating system cluster.

The following operating systems are supported with the installation toolkit in a mixed operating system cluster.

Table 40. Operating systems supported with the installation toolkit in a mixed cluster	
CPU architecture	Operating system
x86_64	<ul style="list-style-type: none"> Red Hat Enterprise Linux 7.x Red Hat Enterprise Linux 8.x SLES 15 Ubuntu 20.04.x and 22.04 ¹
s390x	<ul style="list-style-type: none"> Red Hat Enterprise Linux 7.x Red Hat Enterprise Linux 8.x SLES 15

For latest information about supported operating systems, see [IBM Spectrum Scale FAQ](#) in [IBM Documentation](#).

Note:

- The object protocol is not supported on Red Hat Enterprise Linux 7.x.
- The object protocol is not supported on SLES and Ubuntu.
- The object protocol is not supported on the s390x architecture.

Important:

- For using the installation toolkit in a mixed operating system cluster, all protocol nodes must be running on the same operating system. However, nodes that are running on different minor versions of Red Hat Enterprise Linux 7.x or 8.x can be designated as protocol nodes in the same cluster.
- For using the installation toolkit in a mixed operating system cluster, all nodes must have the same CPU architecture.

The installation toolkit performs the required validations in a mixed operating system cluster to prevent users from attempting any configurations that are not supported. These validations include the following.

<i>Table 41. Validations by the installation toolkit in a mixed operating system cluster</i>	
Operation that is attempted with the installation toolkit	Validations done
Node addition by using the spectrumscale node add <i>NodeName</i> command	Check whether the operating system of the target node is supported. If it is not supported, an error occurs.
Protocol node designation by using the spectrumscale node add <i>NodeName</i> -p command	<ul style="list-style-type: none"> • Check whether the operating system of the target node is SLES 15. If it is SLES 15, a warning is generated that states: Object protocol is not supported on SLES 15. • Check whether all protocol nodes being added are running on the same operating system. If you attempt to add protocol nodes running on different operating systems, then that operation is not allowed. <p>For example, if you have added a protocol node running on Red Hat Enterprise Linux 7.x and then you try to add another protocol node running on SLES 15, then that operation is not allowed.</p>
Protocol enablement by using the spectrumscale enable <i>Protocol</i> command	<p>Check whether the protocol being enabled is supported on the operating system that is running on the node. If the protocol is not supported, an error occurs and you cannot continue with the installation and deployment.</p> <p>For example, if you try to enable object on a node running on SLES 15, an error occurs.</p>

Mixed operating system support during upgrade

During the upgrade precheck, the installation toolkit performs a comparison of the versions of the installed packages with the versions of packages in the repository. In a mixed operating system cluster, the comparison is done with the package repository applicable for the operating system running on the respective nodes.

Note: ¹ On Ubuntu nodes in a mixed operating system cluster, upgrades by using the installation toolkit are not supported. As a workaround, you can use the installation toolkit to upgrade a mixed operating

system cluster with Ubuntu nodes in two hops. For more information, see [“Upgrading mixed operating system cluster with Ubuntu nodes”](#) on page 558.

Preparing to use the installation toolkit

Before you use the installation toolkit, complete the following preparatory steps on all the nodes on which you plan to use the installation toolkit.

During the precheck phase, the installation toolkit check the following items:

- Passwordless SSH is set up.
- Prerequisite kernel packages are installed.
- Needed ports are open.
- Supported OS and architectures are discovered on nodes.
- Base OS repositories are set up.

1. Ensure that the following requirements are met.

Operating systems supported by the installation toolkit

The installation toolkit is supported on the following operating systems.

- Red Hat Enterprise Linux 8.4 and 8.6 operating systems on x86_64, PPC64LE, and s390x architectures.
- Red Hat Enterprise Linux 7.9 operating systems on x86_64, PPC64LE, and s390x architectures.
- SLES 15 operating system on x86_64 and s390x architectures.
- Ubuntu 20.04.x LTS operating system on x86_64 and PPC64LE architectures.
- Ubuntu 22.04 operating system on x86 architecture.

For information about supported operating systems, see [IBM Spectrum Scale FAQ in IBM Documentation](#). Also, check with the operating system vendor to verify that a specific version of the operating system is supported by the vendor.

For information about how the installation toolkit can be used in a cluster that has nodes with mixed operating systems, see [“Mixed operating system support with the installation toolkit”](#) on page 402.

Note:

- The object protocol is not supported on Red Hat Enterprise Linux 7.x.
- The object protocol is not supported on SLES and Ubuntu.
- The object protocol is not supported on the s390x architecture.

Required packages for the installation toolkit

The installation toolkit requires following packages:

- Python 3.6

Note:

- It is recommended that you install Python by using the OS package manager to avoid potential installation issues. For example, **yum install python3**.
- On Ubuntu 20.04.x LTS nodes, you need to install Python 3.6 because it is not included by default in the base Ubuntu 20.04.x LTS operating system. You can install either the dependency package `python` or the minimal python 3.6 environment `python-minimal`.
- If you have Python 2.x and Python 3.x in your environment, ensure that Python 3.x is the default Python version. You can use the **python --version** command to check the default Python version. For example:

```
# python --version
Python 3.6.0
```

- net-tools
- Ansible 2.9.15

Supported Ansible version

The installation toolkit requires Ansible version 2.9.15.

Note: The installation toolkit also works in Ansible 2.10.x environments.

The following Ansible installation considerations apply, depending on your operating system.

Operating system	Ansible installation considerations
Red Hat Enterprise Linux 7.x and 8.x	The installation toolkit installs the supported version of Ansible on the installer node when you run the ./spectrumscale setup -s InstallNodeIP command.
SLES 15	You can use zypper install ansible to install Ansible, but ensure that the Ansible version is 2.9.1 or later.
Ubuntu 20.04.x and 22.04	On Ubuntu nodes, Ansible 2.9.6 might be installed by default or you can install Ansible 2.9.x from the apt repository by using apt-get install ansible=2.9* . You can use the default Ansible version.

- You can manually install Ansible 2.9.15 by using the following command.

```
# pip3 install ansible==2.9.15
```

- You can manually install Ansible 2.10 by using the following command.

```
# pip3 install ansible==2.10
```

For more information, see [Installing Ansible](#).

Note: If the **pip3** command does not work with the installed Python version, use **yum**, **zypper**, or **apt-get** commands.

Root access for the installation toolkit

The installation toolkit must be run as the root user. On Ubuntu, root login might be disabled by default. To use the installation toolkit on Ubuntu, you must enable the root login.

Call home information required for the installation toolkit

The installation toolkit requires call home data to work properly. You must have the *customer name*, the *customer id*, the *customer email*, and the *customer country code* available before installing and configuring call home with the installation toolkit.

Disable auto updates on Ubuntu nodes

On Ubuntu nodes, auto updates or unattended upgrades must be disabled, and kernel auto updates upon node reboot must also be disabled. This must be done to ensure that any updates that are not supported do not get installed. For example, a minor Ubuntu 20.04.x update that is not supported by IBM Spectrum Scale and installation toolkit might get installed if auto updates are not disabled.

Uninstall RPM Package Manager (RPM) on Ubuntu nodes

To use the installation toolkit on Ubuntu nodes, you must uninstall the RPM Package Manager (RPM) from these nodes.

Cluster configuration repository (CCR) is enabled

Before using the installation toolkit, ensure that CCR is enabled. For new installations, CCR is enabled by default. You can check that CCR is enabled in the cluster by using the following command.

```
# mm1scluster

GPFS cluster information
=====
GPFS cluster name:      node1.example.com
GPFS cluster id:       123412789099501234
GPFS UID domain:       node1.example.com
Remote shell command:  sudo wrapper in use
Remote file copy command: sudo wrapper in use
Repository type:       CCR
```

Uninstall Upstream nfs-ganesha package

Before installing IBM Spectrum Scale 5.1.x on Ubuntu operating system, make sure that the upstream NFS Ganesha package is not installed or configured in the protocol nodes. If the NFS Ganesha package is installed or configured, then uninstall the package by using the **apt purge nfs-ganesha** command.

For information on prerequisites for protocols and performance monitoring, see [“Installation prerequisites” on page 352](#).

2. Set up passwordless SSH as follows.

- From the admin node to the other nodes in the cluster.
- From protocol nodes to other nodes in the cluster.
- From every protocol node to rest of the protocol nodes in the cluster.

Note: Passwordless SSH must be set up by using the FQDN and the short name of the node.

The installation toolkit performs verification during the precheck phase to ensure that passwordless SSH is set up correctly. This verification includes:

- Check whether passwordless SSH is set up between all admin nodes and all the other nodes in the cluster. If this check fails, a fatal error occurs.
- Check whether passwordless SSH is set up between all protocol nodes and all the other nodes in the cluster. If this check fails, a warning is displayed.
- Check whether passwordless SSH is set up between all protocol nodes in the cluster. If this check fails, a fatal error occurs.

3. Set up the base OS repositories on your nodes so that package dependencies can be satisfied.

- These package managers must be set up depending on the operating system.
 - Red Hat Enterprise Linux: yum repository must be set up on all nodes in the cluster.

Note: On Red Hat Enterprise Linux 8.x, two package repositories are available: BaseOS and Application Stream (AppStream). To streamline the installation and the upgrade of IBM Spectrum Scale packages on Red Hat Enterprise Linux 8.x nodes, it is recommended to configure the BaseOS and the Application Stream repositories.

- SLES: zypper repository must be set up on all nodes in the cluster.
- Ubuntu: apt repository must be set up on all nodes in the cluster.

Note: Ensure that the **apt-get update** command is working without any errors.

- Configure repositories depending on whether you have internet connection or not. However, ensure that base operating system packages are available and EPEL repositories are disabled.

4. Ensure that the ports that are needed for installation are open.

Note: The installation toolkit checks if the firewall daemon is running and displays a warning if it is running. If the required ports are open, you can ignore the warning.

For information about the ports that need to be open, see *Securing the IBM Spectrum Scale system using firewall* in *IBM Spectrum Scale: Administration Guide*.

5. Ensure that the required kernel packages are installed.

For more information, see [“Software requirements”](#) on page 220.

Note: You might have to force the installation of a specific version of these packages because a package of a version newer than the corresponding kernel might get picked up by default.

6. Ensure that networking is set up in one of the following ways.

- DNS is configured such that all host names, either short or long, are resolvable.
- All host names are resolvable in the `/etc/hosts` file. The host entries in the `/etc/hosts` file are required to be in the following order:

```
<IP address> <FQDN> <Short name>
```

If the entries in the `/etc/hosts` file are not in this order, the installation toolkit displays a warning and continues with the rest of the procedure. The installation toolkit supports both FQDN and short name during installation, upgrade, or deployment. During fresh installations, the nodes are configured based on the entries in `/etc/hosts`. If `/etc/hosts` contains short names then the nodes are added with the short names, if the host names are reachable. For existing clusters, the installation toolkit configures the new node according to the configuration of the existing cluster. For example, if the existing cluster is configured with short names then the new node is added with the short name.

7. Obtain the IBM Spectrum Scale self-extracting installation package from IBM Fix Central.
8. Ensure that the `LC_ALL` and the `LANGUAGE` parameters are set to `en_US.UTF-8`.

You can use the **locale** command to check the current settings. If these parameters are set to a value different than `en_US.UTF-8`, use the following export statements before you issue the `./spectrumscale` commands.

```
export LC_ALL=en_US.UTF-8
export LANGUAGE=en_US.UTF-8
```

You can also add these export statements in the `bash_profile`.

Using the installation toolkit to perform installation tasks: Explanations and examples

Use these explanations and examples of installation toolkit options and tasks to perform installation and deployment by using the installation toolkit.

Note: CES supports HDFS protocols. For installing CES HDFS, the same prerequisites are applicable as for the other protocols as documented in [“Preparing to use the installation toolkit”](#) on page 404 and this section. For specific information about installing CES HDFS, see *Installation* under [CES HDFS](#) in *Big data and analytics support documentation*.

After determining how you want your system to be configured, and after you have reviewed the `spectrumscale` command description in the *Command reference* section *IBM Spectrum Scale: Command and Programming Reference*, you can tailor these examples to do the following tasks:

- Set up your installer node.
- Add node and NSD specifications, file system information, and configuration properties to your cluster definition file, and then install IBM Spectrum Scale and configure your cluster according to the information in that file.
- Specify environment details, identify which protocols are to be enabled, and define protocol-specific properties in the cluster definition file, and then deploy protocols on your system.

For information on using the installation toolkit for the installation and the deployment of protocol and client nodes in a cluster containing ESS, see [“ESS awareness with the installation toolkit”](#) on page 429.

Before using the installation toolkit for installation and deployment, ensure that the preparatory steps are completed. For more information, see [“Preparing to use the installation toolkit” on page 404](#).

Using the installation toolkit to perform installation and deployment involves doing these tasks.

1. [“Setting up the installer node” on page 408](#)
2. [“Defining the cluster topology for the installation toolkit” on page 409](#)
3. [“Setting configuration parameters before installation” on page 413](#)
4. [“Installing IBM Spectrum Scale and creating a cluster” on page 414](#)
5. [“Deploying protocols” on page 416](#)

For more installation toolkit related information, see the following topics.

- [“Installation of performance monitoring tool using the installation toolkit” on page 420](#)
- [“Enabling and configuring file audit logging using the installation toolkit” on page 422](#)
- [“Enabling and configuring call home using the installation toolkit” on page 424](#)
- [“Installing IBM Spectrum Scale management GUI by using the installation toolkit” on page 426](#)
- [“Logging and debugging for installation toolkit” on page 421](#)

Setting up the installer node

The first step in using the installation toolkit for installing IBM Spectrum Scale and deploying protocols is to configure your installer node.

A good candidate for your installer node is the GPFS admin node because a prerequisite of this node is that it must be able to communicate with all other nodes without the need to provide a password. Also, SSH connections between the admin node and all other nodes must be set up to suppress prompting. Therefore, no prompts for password must display when you are using SSH among any cluster nodes to and from each other, and to and from the server.

Note: Before you configure a SLES or an Ubuntu node as the installer node, ensure that the supported version of Ansible is installed on that node. On Red Hat Enterprise Linux nodes, the installation toolkit installs the supported version of Ansible. For more information, see [“Supported Ansible version” on page 405](#).

Tip: Ensure that there is at least 5 GB of free space on the node to download, extract, and install the packages.

1. Change the directory to where the installation toolkit is extracted. The default extraction path for 5.1.5.x is as follows. This path varies depending on the version.

```
cd /usr/lpp/mmfs/5.1.5.x/ansible-toolkit
```

2. Configure the installer node by issuing the following command.

```
./spectrumscale setup -s InstallNodeIP -i SSHIdentity
```

The `-s` argument identifies the IP address that nodes use to retrieve their configuration. This IP address is associated with a device on the installer node. This validation is automatically done during the setup phase.

Note: If there are multiple subnets in your cluster, it is recommended to specify a private IP address with the `./spectrumscale setup -s` command.

Optionally, you can specify a private SSH key to be used to communicate with nodes in the cluster definition file, by using the `-i` argument.

In an Elastic Storage Server (ESS) cluster, if you want to use the installation toolkit to install IBM Spectrum Scale and deploy protocols, you must specify the setup type as `ess` while setting up the installer node as follows.

```
./spectrumscale setup -s InstallNodeIP -i SSHIdentity -st ess
```

For more information, see [“ESS awareness with the installation toolkit” on page 429](#).

If you want to use the installation toolkit to install IBM Spectrum Scale Erasure Code Edition, you must specify the setup type as `ece` while setting up the installer node as follows.

```
./spectrumscale setup -s InstallNodeIP -i SSHIdentity -st ece
```

For more information, see IBM Spectrum Scale Erasure Code Edition documentation.

Defining the cluster topology for the installation toolkit

Use these instructions to set up the cluster definition file before installing IBM Spectrum Scale and deploying protocols.

1. [“Adding node definitions to the cluster definition file” on page 409](#)
2. [“Adding and configuring NSD server nodes in the cluster definition file” on page 411](#)
3. [“Defining file systems” on page 412](#)

Adding node definitions to the cluster definition file

You can add node definitions to the cluster definition file by using the **./spectrumscale node add** command.

Note: For information on adding protocol nodes to the cluster definition file, see [“Adding protocol nodes to the cluster definition file” on page 417](#).

1. If the installation toolkit is being used from a location outside of any of the nodes to be installed, an admin node is required. The admin node is used to run cluster-wide commands.

To specify an admin node in the cluster definition file, use the `-a` argument.

```
./spectrumscale node add gpfsnode1 -a
```

If no admin node is specified in the cluster definition file, the node on which the installation toolkit is running is automatically designated as the admin node. If GUI nodes are to be installed, each GUI node must also be marked as an admin node.

The role of an admin node is to serve as the coordinator of the installation, deployment, and upgrade when using the installation toolkit. This node also acts as a central repository for all IBM Spectrum Scale packages. For larger clusters, it is important to have an admin node with plenty of network bandwidth to all other nodes in the cluster.

2. To add client nodes to the cluster definition file, provide no arguments.

```
./spectrumscale node add gpfsnode1
```

3. To add manager nodes to the cluster definition file, use the `-m` argument.

```
./spectrumscale node add gpfsnode2 -m
```

If no manager nodes are added to the cluster definition, the installation toolkit automatically designates manager nodes using the following algorithm:

- a. First, all protocol nodes in the cluster definition are designated as manager nodes.
- b. If there are no protocol nodes, all NSD nodes in the cluster definition are designated as manager nodes.

- c. If there are no NSD nodes, all nodes in the cluster definition are designated as manager nodes.
- 4. To add quorum nodes to the cluster definition, use the `-q` argument.

```
./spectrumscale node add gpfsnode3 -q
```

If no quorum nodes are added to the cluster definition, the installation toolkit automatically designates quorum nodes using the following algorithm:

- a. If the number of nodes in the cluster definition is less than 4, all nodes are designated as quorum nodes.
- b. If the number of nodes in the cluster definition is between 4 and 9 inclusive, 3 nodes are designated as quorum nodes.
- c. If the number of nodes in the cluster definition is between 10 and 18 inclusive, 5 nodes are designated as quorum nodes.
- d. If the number of nodes in the cluster definition is greater than 18, 7 nodes are designated as quorum nodes.

This algorithm preferentially selects NSD nodes as quorum nodes. If the number of NSD nodes is less than the number of quorum nodes to be designated then any other nodes are selected until the number of quorum nodes is satisfied.

- 5. To add NSD servers to the cluster definition, use the `-n` argument.

```
./spectrumscale node add gpfsnode4 -n
```

- 6. To add Graphical User Interface servers to the cluster definition, use the `-g` argument. A GUI server must also be an admin node.

```
./spectrumscale node add gpfsnode3 -g -a
```

If no nodes have been specified as management GUI servers, then the GUI is not installed. It is recommended to have at least 2 management GUI interface servers and a maximum of 3 for redundancy.

- 7. To add a call home node to the cluster definition, use the `-c` argument.

```
./spectrumscale node add CallHomeNode -c
```

If a call home node is not specified, the installation toolkit assigns one of the nodes in the cluster as the call home node.

- 8. To display a list of all nodes in the cluster definition file, use the **`./spectrumscale node list`** command. For example:

```
./spectrumscale node list
```

```
[ INFO ] List of nodes in current configuration:
[ INFO ] [Installer Node]
[ INFO ] 192.0.2.1
[ INFO ] [Cluster Name]
[ INFO ] gpfscluster01
[ INFO ]
[ INFO ] GPFS Node      Admin  Quorum  Manager  NSD Server  Protocol  GUI Server  OS
Arch
[ INFO ] gpfsnode1      X      X              X              rhel7
x86_64
[ INFO ] gpfsnode2              X              rhel7
x86_64
[ INFO ] gpfsnode3      X      X      X      X              X      rhel7
x86_64
[ INFO ] gpfsnode4              X      X      X              rhel7
x86_64
```

- If you want to use the installation toolkit to install IBM Spectrum Scale and deploy protocols in an Elastic Storage Server (ESS) cluster, you must add the EMS node of the ESS system as follows.

```
./spectrumscale node add -e EMSNode
```

For more information, see [“ESS awareness with the installation toolkit”](#) on page 429.

- For information on adding nodes to an existing installation, see [“Adding nodes, NSDs, or file systems to an existing cluster”](#) on page 443.

Adding and configuring NSD server nodes in the cluster definition file

Note: A CES shared root file system is required for protocols deployment with IBM Spectrum Scale.

1. To configure NSDs, you must first add your NSD server nodes to the configuration:

```
./spectrumscale node add -n nsdserver1  
./spectrumscale node add -n nsdserver2  
./spectrumscale node add -n nsdserver3
```

2. Once NSD server nodes are in the configuration, add NSDs to the configuration:

```
./spectrumscale nsd add /dev/sdb -p nsdserver1 -s nsdserver2,nsdserver3,...
```

- The installation toolkit supports standalone NSDs which connect to a single NSD server or shared NSDs which connect to both a primary and a secondary NSD server.
- When adding a standalone NSD, skip the secondary NSD server parameter.
- When adding a shared NSD, it is important to know the device name on the node which is to become the primary NSD server. It is not necessary to know the device name on the secondary NSD server because the device is looked up using its UUID.

Note: Although it is not necessary to know the device name on the secondary NSD server, it may be helpful to create a consistent mapping of device names if you are using multipath. For more information, see [“NSD disk discovery”](#) on page 23.

Here is an example of adding a shared NSD to the configuration by specifying the device name on the primary server along with the primary and secondary servers.

3. The name of the NSD is automatically generated based on the NSD server names. This can be changed after the NSD has been added by using the modify command and specifying a new name with the `-n` flag; the new name must be unique:

```
./spectrumscale nsd modify nsd_old_name -n nsd_new_name
```

4. It is possible to view all NSDs currently in the configuration using the list command:

```
./spectrumscale nsd list
```

5. To remove a single NSD from the configuration, supply the name of the NSD to the delete command:

```
./spectrumscale nsd delete nsd_name
```

6. To clear all NSDs and start from scratch, use the clear command:

```
./spectrumscale nsd clear
```

7. Where multiple devices are connected to the same pair of NSD servers, they can be added in bulk either by providing a list of all devices, or by using wild cards:

```
./spectrumscale nsd add -p nsdserver1 -s nsdserver2 /dev/dm-1 /dev/dm-2 /dev/dm-3
```

or

```
./spectrumscale nsd add -p nsdserver1 -s nsdserver2 "/dev/dm-*)"
```

A connection is made to the primary server to expand any wild cards and check that all devices are present. When using wild cards, it is important to ensure that they are properly escaped, as otherwise they may be expanded locally by your shell. If any devices listed cannot be located on the primary server, a warning is displayed, but the command continues to add all other NSDs.

- When adding NSDs, it is good practice to have them distributed such that each pair of NSD servers is equally loaded. This is usually done by using one server as a primary for half of the NSDs, and the other server as primary for the remainder.

```
./spectrumscale nsd add "/dev/dm-*)" -p serverA -s serverB
```

```
[ INFO ] Connecting to serverA to check devices and expand wildcards.
[ INFO ] Adding NSD serverA_serverB_1 on serverA using device /dev/dm-0.
[ INFO ] Adding NSD serverA_serverB_2 on serverA using device /dev/dm-1.
$ ./spectrumscale nsd list
[ INFO ] Name                               FS      Size(GB) Usage   FG Pool   Device
Servers
[ INFO ] serverA_serverB_1 Default 13      Default 1 Default /dev/dm-0 serverA,serverB
[ INFO ] serverA_serverB_2 Default 1       Default 1 Default /dev/dm-1 serverA,serverB
```

- Ordinarily a connection is made to the primary NSD server when adding an NSD. This is done to check device names and so that details such as the disk size can be determined, but is not vital. If it is not feasible to have a connection to the nodes while adding NSDs to the configuration, these connections can be disabled using the `--no-check` flag. Extra care is needed to manually check the configuration when using this flag.

```
./spectrumscale nsd add /dev/sda -p nsdserver1 -s nsdserver2 --no-check
```

- You can set the failure group, file system, pool, and usage of an NSD in two ways:

- using the add command to set them for multiple new NSDs at once
- using the modify command to modify one NSD at a time

```
./spectrumscale nsd add "/dev/dm-*)" -p nsdserver1 -s nsdserver2 \
-po pool1 -u dataOnly -fg 1 -fs filesystem_1
./spectrumscale nsd modify nsd_name -u metadataOnly -fs filesystem_1
```

For information on adding NSDs to an existing installation, see [“Adding nodes, NSDs, or file systems to an existing cluster”](#) on page 443.

Defining file systems

File systems are defined with the NSD configuration and they are only created at the time of installation, if there are NSDs assigned to them.

Note: A CES shared root file system is required for protocols deployment with IBM Spectrum Scale.

- To specify a file system, use the **nsd add** or **nsd modify** command to set the file system property of the NSD:

```
./spectrumscale nsd add "/dev/dm-*)" -p server1 -s server2 -fs filesystem_1
[ INFO ] The installer will create the new file system filesystem_1 if it does not already exist.
```

```
./spectrumscale nsd modify server1_server2_1 -fs filesystem_2
[ INFO ] The installer will create the new file system filesystem_2 if it does not already exist.
```

2. To list all file systems that currently have NSDs assigned to them, use the **list** command. This also displays file system properties including the block size and mount point:

```
./spectrumscale filesystem list
[ INFO ] Name      BlockSize  Mountpoint      NSDs Assigned
[ INFO ] filesystem_1 Default    /ibm/filesystem_1 3
[ INFO ] filesystem_2 Default    /ibm/filesystem_2 1
```

3. To alter the block size and mount point from their default values, use the modify command:

```
./spectrumscale filesystem modify filesystem_1 -B 1M -m /gpfs/gpfs0
```

Important: NSDs and file systems are created when the **./spectrumscale install** command is issued.

It is not possible to directly rename or delete a file system; this is instead done by reassigning the NSDs to a different file system using the **nsd modify** command.

At this point, the cluster definition file contains:

- Nodes and node types defined
- NSDs optionally defined
- File systems optionally defined

To proceed with the IBM Spectrum Scale installation, go to the next task: [“Installing IBM Spectrum Scale and creating a cluster” on page 414.](#)

For information on adding file systems to an existing installation, see [“Adding nodes, NSDs, or file systems to an existing cluster” on page 443.](#)

Setting configuration parameters before installation

After you define nodes in the cluster definition file, you can set certain configuration parameters in the cluster definition. Use the **./spectrumscale config gpfs** command for this purpose.

- To specify a cluster name in the cluster definition, use the **-c** argument.

```
./spectrumscale config gpfs -c gpfscluster01.my.domain.name.com
```

In this example, `gpfscluster01.my.domain.name.com` is the cluster name.

If no cluster name is specified, the GPFS admin node name is used as the cluster name. If the user-provided name contains periods, it is assumed to be a fully qualified domain name. If the cluster name is not a fully qualified domain name, the cluster name domain name is inherited from the admin node domain name.

- To specify a profile in the cluster definition file to be set on cluster creation, use the **-p** argument.

```
./spectrumscale config gpfs -p randomio
```

The valid values for **-p** option are `default` (for `gpfsProtocolDefaults` profile), `randomio` (for `gpfsProtocolRandomIO` profile), and *ProfileName* (for user-defined profile).

- The system-defined profiles are based on workload type: sequential I/O (`gpfsProtocolDefaults`) or random I/O (`gpfsProtocolRandomIO`). These defined profiles can be used to provide initial default tunables or settings for a cluster. If more tunable changes are required, see `mmchconfig` command and the `mmcrcluster` command in *IBM Spectrum Scale: Command and Programming Reference*.
- You can specify a user-defined profile of attributes to be applied. The profile file specifies GPFS configuration parameters with values different than the documented defaults.

A user-defined profile must have the following properties:

- It must not begin with the string `gpfs`.
- It must have the `.profile` suffix.

- It must be located in the `/var/mmfs/etc/` directory.

Note: The installation toolkit places the user-defined profile in the `/var/mmfs/etc/` from the path that you specify with the `./spectrumscale config gpfs -p PathToUserDefinedProfile` command.

A sample user-defined profile is available at this path: `/usr/lpp/mmfs/samples/sample.profile`. For more information, see *mmcrcluster command* and *mmchconfig command* in *IBM Spectrum Scale: Command and Programming Reference*.

If no profile is specified in the cluster definition, the `gpfsProtocolDefaults` profile is automatically set on cluster creation.

- To specify the remote shell binary to be used by GPFS, use the `-r` argument.

```
./spectrumscale config gpfs -r /usr/bin/ssh
```

If no remote shell is specified in the cluster definition, `/usr/bin/ssh` is used as the default.

- To specify the remote file copy binary to be used by GPFS, use the `-rc` argument.

```
./spectrumscale config gpfs -rc /usr/bin/scp
```

If no remote file copy binary is specified in the cluster definition, `/usr/bin/scp` is used as the default.

- To specify an ephemeral port range to be set on all GPFS nodes, use the `-e` argument.

```
./spectrumscale config gpfs -e 61000-62000
```

For information about the ephemeral port range, see *GPFS port usage* in *IBM Spectrum Scale: Administration Guide*.

If no port range is specified in the cluster definition, 60000-61000 is used as default.

- To view the current GPFS configuration settings, issue the following command.

```
./spectrumscale config gpfs --list
```

```
[ INFO ] No changes made. Current settings are as follows:
[ INFO ] GPFS cluster name is gpfscluster01
[ INFO ] GPFS profile is default
[ INFO ] Remote shell command is /usr/bin/ssh
[ INFO ] Remote file copy command is /usr/bin/scp
[ INFO ] GPFS Daemon communication port range is 61000-62000
```

Installing IBM Spectrum Scale and creating a cluster

After you set up the installer node, and define nodes, NSDs, and file systems in the cluster definition file, and set certain configuration parameters, you can install IBM Spectrum Scale according to the topology defined in the cluster definition file.

1. If call home is enabled in the cluster definition file, specify the minimum call home configuration parameters.

```
./spectrumscale callhome config -n CustName -i CustID -e CustEmail -cn CustCountry
```

For more information, see [“Enabling and configuring call home using the installation toolkit”](#) on page 424.

2. Do environment checks before initiating the installation procedure.

```
./spectrumscale install -pr
```

This step is not mandatory because running `./spectrumscale install` with no arguments also runs these checks before the installation.

3. Start the IBM Spectrum Scale installation and the creation of the cluster.

```
./spectrumscale install
```

Understanding what the installation toolkit does during the installation

- If the installation toolkit is being used to install IBM Spectrum Scale on all nodes, create a new cluster, and create NSDs, it automatically does the following outlined steps.
- If the installation toolkit is being used to add nodes to an existing GPFS cluster and create new NSDs, it automatically does the following steps.
- If all nodes in the cluster definition file are in a cluster, then the installation toolkit automatically does the following steps.

To add nodes to an existing GPFS cluster, at least one node in the cluster definition must belong to the cluster where the nodes not in a cluster are to be added. The cluster name in the cluster definition must also exactly match the cluster name outputted by **mmlscluster**.

When you run the **./spectrumscale install** command, the installation toolkit does these steps:

Install IBM Spectrum Scale on all nodes, create a new GPFS cluster, and create NSDs

- Run preinstall environment checks
- Install the IBM Spectrum Scale packages on all nodes
- Build the GPFS portability layer on all nodes
- Install and configure performance monitoring tools
- Create a GPFS cluster
- Configure licenses
- Set ephemeral port range
- Create NSDs, if any are defined in the cluster definition
- Create file systems, if any are defined in the cluster definition
- Run post-install environment checks

Add nodes to an existing GPFS cluster and create any new NSDs

- Run preinstall environment checks
- Install the IBM Spectrum Scale packages on nodes to be added to the cluster
- Install and configure performance monitoring tools on nodes to be added to the cluster
- Add nodes to the GPFS cluster
- Configure licenses
- Create NSDs, if any are defined in the cluster definition
- Create file systems, if any are defined in the cluster definition
- Run post-installation environment checks

Important: The installation toolkit does not alter anything on the existing nodes in the cluster. You can determine the existing nodes in the cluster by using the **mmlscluster** command.

The installation toolkit might change the performance monitoring collector configuration if you are adding the new node as a GUI node or an NSD node, due to collector node prioritization. However, if you do not want to change the collector configuration then you can use the **./spectrumscale config perfmon -r off** command to disable performance monitoring before initiating the installation procedure.

All nodes in the cluster definition are in a cluster

- Run preinstall environment checks
- Skip all steps until NSD creation

- Create NSDs (if any new NSDs are defined in the cluster definition)
- Run post-install environment checks

For more information, see **spectrumscale command** in *IBM Spectrum Scale: Command and Programming Reference*.

What to do next

Upon completion of the installation, you have an active GPFS cluster. Within the cluster, NSDs might be created, file systems might be created, performance monitoring is configured, and all product licenses are accepted.

The installation can be rerun in the future to:

- Add NSD server nodes
- Add GPFS client nodes
- Add GUI nodes
- Add NSDs
- Define new file systems

Deploying protocols

Use this information to deploy protocols in an IBM Spectrum Scale cluster using the installation toolkit.

Deployment of protocol services is performed on a subset of the cluster nodes that are designated as protocol nodes using the **./spectrumscale node add node_name -p** command. Protocol nodes have an additional set of packages installed that allow them to run the NFS, SMB, and Object protocol services.

Data is served through these protocols from a pool of addresses designated as Export IP addresses or CES "public" IP addresses using **./spectrumscale config protocols -e IP1,IP2,IP3...** or added manually using the **mmces address add** command. The allocation of addresses in this pool is managed by the cluster, and IP addresses are automatically migrated to other available protocol nodes in the event of a node failure.

Before deploying protocols, there must be a GPFS cluster that has GPFS started and it has at least one file system for the CES shared root file system.

Notes: Here are a few considerations for deploying protocols.

1. All the protocol nodes must be running the supported operating systems, and the protocol nodes must have the same CPU architecture. Although the other nodes in the cluster could be on other platforms and operating systems.

For information about supported operating systems for protocol nodes and their required minimum kernel levels, see [IBM Spectrum Scale FAQ in IBM Documentation](#).

2. The packages for all protocols are installed on every node designated as a protocol node; this is done even if a service is not enabled in your configuration.
3. Services are enabled and disabled cluster wide; this means that every protocol node serves all enabled protocols.
4. If SMB is enabled, the number of protocol nodes is limited to 16 nodes.
5. If your protocol node has Red Hat Enterprise Linux 7.x installed, there might be an NFS service already running on the node that can cause issues with the installation of IBM Spectrum Scale NFS packages. To avoid these issues, before starting the deployment, you must do the following:
 - a. Stop the NFS service using the **systemctl stop nfs.service** command.
 - b. Disable the NFS service using the **systemctl disable nfs.service** command.

This command ensures that this change is persistent after system reboot.

6. The installation toolkit does not support adding protocol nodes to an existing ESS cluster prior to ESS version 3.5.

Deploying protocols involves doing the following sub-tasks.

1. [“Defining a shared file system for protocols” on page 417](#)
2. [“Adding protocol nodes to the cluster definition file” on page 417](#)
3. [“Enabling NFS and SMB” on page 417](#)
4. [“Configuring object” on page 418](#)
5. [“Adding CES IP addresses” on page 419](#)
6. [“Running the ./spectrumscale deploy command” on page 420](#)

Defining a shared file system for protocols

To use protocol services, a shared file system (CES shared root file system) must be defined. If GPFS has already been configured, the shared file system can be specified manually or by re-running the **spectrumscale install** command to assign an existing NSD to the file system. If re-running **spectrumscale install**, be sure that your NSD servers are compatible with the installation toolkit and contained within the cluster definition file.

You can use the **./spectrumscale config protocols** command to define the shared file system with the **-f** option and the shared file system mount point or path with the **-m** option.

For example: **./spectrumscale config protocols -f ceshared -m /gpfs/ceshared.**

To view the current settings, issue this command:

```
./spectrumscale config protocols --list
```

```
[ INFO ] No changes made. Current settings are as follows:
[ INFO ] Shared File System Name is ceshared
[ INFO ] Shared File System Mountpoint or Path is /gpfs/ceshared
```

Adding protocol nodes to the cluster definition file

To deploy protocols on nodes in your cluster, they must be added to the cluster definition file as protocol nodes.

Issue the following command to designate a node as a protocol node:

```
./spectrumscale node add NODE_IP -p
```

Enabling NFS and SMB

To enable or disable a set of protocols, use the **./spectrumscale enable** and **./spectrumscale disable** commands. For example:

```
./spectrumscale enable smb nfs
```

```
[ INFO ] Enabling SMB on all protocol nodes.
[ INFO ] Enabling NFS on all protocol nodes.
```

You can view the current list of enabled protocols by using the **spectrumscale node list** command. For example:

```
./spectrumscale node list
```

```
[ INFO ] List of nodes in current configuration:
[ INFO ] [Installer Node]
[ INFO ] 192.0.2.1
[ INFO ]
[ INFO ] [Cluster Name]
[ INFO ] ESDev1
```

```

[ INFO ]
[ INFO ] [Protocols]
[ INFO ] Object : Disabled
[ INFO ] SMB : Enabled
[ INFO ] NFS : Enabled
[ INFO ]
[ INFO ] GPFS Node
OS Arch Admin Quorum Manager NSD Server Protocol GUI Server
[ INFO ] ESDev1-GPFS1 X X X X
rhel7 x86_64
[ INFO ] ESDev1-GPFS2 X
rhel7 x86_64
[ INFO ] ESDev1-GPFS3 X
rhel7 x86_64
[ INFO ] ESDev1-GPFS4 X X X X
rhel7 x86_64
[ INFO ] ESDev1-GPFS5 X X X X
rhel7 x86_64

```

Configuring object



Attention:

- The object protocol is not supported in IBM Spectrum Scale 5.1.0.0. If you want to deploy object, install the IBM Spectrum Scale 5.1.0.1 or a later release.
- If SELinux is disabled during installation of IBM Spectrum Scale for object storage, enabling SELinux after installation is not supported.

If the object protocol is enabled, further protocol-specific configuration is required. You can configure these options by using the **spectrumscale config object** command, which has the following parameters:

```

usage: spectrumscale config object [-h] [-l] [-f FILESYSTEM] [-m MOUNTPPOINT]
                                   [-e ENDPOINT] [-o OBJECTBASE]
                                   [-i INODEALLOCATION]
                                   [-au ADMINUSER] [-ap ADMINPASSWORD]
                                   [-SU SWIFTUSER] [-sp SWIFTPASSWORD]
                                   [-dp DATABASEPASSWORD]
                                   [-s3 {on,off}]

```

The object protocol requires a dedicated fileset as its back-end storage; this fileset is defined using the **--filesystem (-f)**, **--mountpoint (-m)** and **--objectbase (-o)** flags to define the file system, mount point, and fileset respectively.

The **--endpoint (-e)** option specifies the host name that is used for access to the file store. This should be a round-robin DNS entry that maps to all CES IP addresses; this distributes the load of all keystone and object traffic that is routed to this host name. Therefore, the endpoint is an IP address in a DNS or in a load balancer that maps to a group of export IPs (that is, CES IPs that were assigned on the protocol nodes).

The following user name and password options specify the credentials used for the creation of an admin user within Keystone for object and container access. The system prompts for these during **./spectrumscale deploy** pre-check and **./spectrumscale deploy** run if they have not already been configured. The following example shows how to configure these options to associate user names and passwords: **./spectrumscale config object -au -admin -ap -dp**

The **-ADMINUSER (-au)** option specifies the admin user name. This credential is for the Keystone administrator. This user can be local or on remote authentication server based on authentication type used.

The **-ADMINPASSWORD (-ap)** option specifies the password for the admin user.

Note: You are prompted to enter a Secret Encryption Key which is used to securely store the password. Choose a memorable pass phrase which you are prompted for each time you enter the password.

The **-SWIFTUSER (-su)** option specifies the Swift user name. The **-ADMINUSER (-au)** option specifies the admin user name. This credential is for the Swift services administrator. All Swift services are run in this

user's context. This user can be local or on remote authentication server based on authentication type used.

The `-SWIFTPASSWORD(-sp)` option specifies the password for the Swift user.

Note: You are prompted to enter a Secret Encryption Key which is used to securely store the password. Choose a memorable pass phrase which you are prompted for each time you enter the password.

The `-DATABASEPASSWORD(-dp)` option specifies the password for the object database.

Note: You are prompted to enter a Secret Encryption Key which is used to securely store the password. Choose a memorable pass phrase which you are prompted for each time you enter the password.

The `-s3` option specifies whether the S3 (Amazon Simple Storage Service) API should be enabled.

Adding CES IP addresses

Note: This is mandatory for protocol deployment.

CES "public" IP addresses or export IP addresses are used to export data through the protocols (NFS, SMB, object). File and object clients use these public IPs to access data on GPFS file systems. Export IP addresses are shared between all protocols and they are organized in a public IP address pool; there can be fewer public IP addresses than protocol nodes. Export IP addresses must have an associated host name and reverse DNS lookup must be configured for each export IP address.

1. Add export IP addresses to your cluster by using this command:

```
./spectrumscale config protocols -e EXPORT_IP_POOL
```

Where `EXPORT_IP_POOL` is a comma-separated list of IP addresses.

In the CES interface mode, you must specify the CES IP addresses with the installation toolkit in the Classless Inter-Domain Routing (CIDR) notation. In the CIDR notation, the IP address is followed by a forward slash and the prefix length.

```
IPAddress/PrefixLength
```

For example,

IPv6

```
2001:0DB8::/32
```

IPv4

```
192.0.2.0/20
```

You must specify the prefix length for every CES IP address, otherwise you cannot add the IP address by using the installation toolkit.

- When you are using IPv6 addresses, prefix length must be in the range 1 - 124.
- When you are using IPv4 addresses, prefix length must be in the range 1 - 30.

2. If you are using the CES interface mode, specify the interfaces by using the following command.

```
./spectrumscale config protocols -i INTERFACES
```

Where `INTERFACES` is the comma-separated list of network interfaces. For example, `eth0,eth1`.

3. View the current configuration by using the following command:

```
./spectrumscale node list
```

View the CES shared root and the IP address pool by using the following command:

```
./spectrumscale config protocols -l
```

View the object configuration by using the following command:

```
./spectrumscale config object -l
```

Running the `./spectrumscale deploy` command

After adding the previously-described protocol-related definition and configuration information to the cluster definition file, you can deploy the protocols specified in that file.

You can also use the **mmnetverify** command to identify any network problems before doing the deployment. For more information, see *mmnetverify command* in *IBM Spectrum Scale: Command and Programming Reference*.

Use the following command to deploy protocols:

```
./spectrumscale deploy
```

Note: You are prompted for the Secret Encryption Key that you provided while configuring object and/or authentication unless you disabled prompting.

This does the following:

- Performs pre-deploy checks.
- Deploys protocols as specified in the cluster definition file.
- Performs post-deploy checks.

You can explicitly specify the `--precheck (-pr)` option to perform a dry run of pre-deploy checks without starting the deployment. This is not required, however, because **./spectrumscale deploy** with no argument also runs these checks. Alternatively, you can specify the `--postcheck (-po)` option to perform a dry run of post-deploy checks without starting the deployment. These options are mutually exclusive.

After a successful deployment, you can verify the cluster and CES configuration by running this command:

```
$ /usr/lpp/mmfs/bin/mmcluster --ces
```

What to do next

You can rerun the **./spectrumscale deploy** command in the future to do the following tasks:

- Add protocol nodes
- Enable additional protocols

Installation of performance monitoring tool using the installation toolkit

By default, the performance monitoring tool gets installed when IBM Spectrum Scale is installed on the system using the `./spectrumscale install` command.

For more information on the performance monitoring tool, see *Using the performance monitoring tool* in *IBM Spectrum Scale: Problem Determination Guide*.

During the installation, deployment, and upgrade phases, the installation toolkit performs following performance monitoring related configuration changes.

Installation

- Sets nodes to performance monitoring nodes (**mmchnode --perfmon -N Node1,Node2,Node3**)
- Installs `pmsensor` on any nodes that do not have the sensor.
- Installs `pmcollector` on 2 nodes. Preference is the GUI node first, followed by NSD nodes.
- Starts all sensors or collectors and activates all the default file system or node sensors.
- Resets `GPFSDiskCap` and `GPFSFilesetQuota` sensors to default values: 86400 and 3600 respectively and restricts them to the 1st collector node.

Deployment

- Sets nodes to performance monitoring nodes (**mmchnode --perfmon -N Node1,Node2,Node3**)
- Installs **pmsensor** on any nodes that do not have the sensor.
- Installs **pmcollector** on 2 nodes. Preference is the GUI node first, followed by NSD nodes.
- Starts all sensors or collectors and activates all the default file system or node sensors.
- Resets **GPFSDiskCap** and **GPFSFilesetQuota** sensors to default values: 86400 and 3600 respectively and restricts them to the 1st collector node.
- Adds protocol sensors, corresponding to the activated protocols, to the active performance monitoring configuration and restricts them to protocol nodes. These sensors are **NFSIO**, **SMBStats**, **SMBGlobalStats**, **CTDBStats**, **CTDBDBStats**, **SwiftAccount**, **SwiftContainer**, **SwiftObject**, and **SwiftProxy**.

Upgrade

- Upgrades **pmsensor** and **pmcollector**.
- Resets **GPFSDiskCap** and **GPFSFilesetQuota** sensors to default values: 86400 and 3600 respectively and restricts them to the 1st collector node.

Note: The installation toolkit does not set any **NFSIO** parameter related setting.

During the installation toolkit prechecks if it is found that the collector nodes require reconfiguring then the **./spectrumscale config perfmon -r on** command needs to be run to enable the installation toolkit to perform this reconfiguration.

Note: A reconfiguration results in the removal of an existing collector and it might move the collector to different nodes and reset sensor data. Custom sensors and data might be erased. To disable reconfiguration, run the **./spectrumscale config perfmon -r off** command.

If reconfiguration is disabled then the installation toolkit performs all installation tasks except for those related to performance monitoring and the management GUI, if GUI servers are specified.

After the installation or the deployment completes and sensors are added, the latest state of events of type **TIPS** might not get immediately refreshed. As a result, the output of the **mmhealth node show ces** command might contain messages such as the following for the **CES** component:

```
nfs_sensors_not_configured(NFSIO)
smb_sensors_not_configured(SMBGlobalStats, SMBStats)
```

To resolve this issue, run the **mmhealth node show --refresh** command.

For information on how to install the performance monitoring tool manually, see [“Manually installing the performance monitoring tool”](#) on page 376.

For information on performance monitoring related configuration in a cluster containing **ESS**, see [“ESS awareness with the installation toolkit”](#) on page 429.

For information on configuring the performance monitoring tool, see *Configuring the Performance Monitoring tool* in *IBM Spectrum Scale: Problem Determination Guide*.

Logging and debugging for installation toolkit

The installation toolkit reports progress to the console, and certain options also log output to a file.

Console output

The installation toolkit reports progress to the console. By default, only information level messages are displayed. If you need more verbose output, pass the **-v** flag to the **spectrumscale** command.

Note: This flag must be the first argument passed to the **spectrumscale** command. For example:

```
./spectrumscale -v install --precheck
```

Log files

In addition to console output, the `install` and `deploy` functions of the installation toolkit log verbose output to the `logs` subdirectory of the installation toolkit: `/usr/lpp/mmfs/x.x.x.x/ansible-toolkit/logs`

The full log path is also displayed on the console when logging begins. The log file contains verbose output even if the `-v` flag is not specified on the command line.

Installation and deployment checks

The `install` and `deploy` functions have a set of checks performed before and after their main action. These checks can be triggered separately by using either the `--precheck` or `--postcheck` flag.

It is safe to run the pre- and post-checks multiple times, because during these checks cluster data is obtained but no changes are done.

Note: As with the full `install` and `deploy` functions, the pre- and post-checks create a detailed log file that can be used to provide further details about errors encountered.

Gathering support information

If an unresolvable error is encountered when using the installation toolkit, a support package can be gathered by using the `installer.snap.py` script that is available in the same directory as the installation toolkit.

Note: The `installer.snap.py` script must be run from the installation toolkit directory.

The `installer.snap.py` script gathers information from each of the nodes defined in the configuration and packages it such that it is ready for submission to the IBM Support Center. It creates a tar file with the prefix `installer_snap` in the `/tmp` directory.

The script gathers the following information:

- Cluster configuration file
- installation toolkit logs
- Information from the local node of the protocol services
- A GPFS snap file from each node defined in the configuration
- Basic environment information from the local node including:
 - Current OS and kernel version
 - System message log (`dmesg`)
 - File system usage
 - Network configuration
 - Current processes

Collection of core dump data

For information about changing configuration to enable collection of core dump data from protocol nodes, see *Configuration changes required on protocol nodes to collect core dump data* in *IBM Spectrum Scale: Problem Determination Guide*.

Enabling and configuring file audit logging using the installation toolkit

You can use the installation toolkit to enable and configure the file audit logging function in the cluster definition file. After enabling this function at the cluster level, you must enable it on file systems. The file audit logging package (`gpfs.librdkafka`) is installed on all supported nodes in the cluster specified to the installation toolkit during the installation, even if file audit logging is not enabled in the cluster configuration. In a cluster containing an ESS system wherein the setup type is `ESS` or `ess` in the cluster definition file, the file audit logging packages are installed on protocol nodes and client nodes. They are

not installed on ESS EMS and I/O server nodes. Based on the file audit logging configuration options specified in the cluster definition file using the installation toolkit, the function is enabled and configured in the cluster accordingly during the deployment.

For information on required packages for file audit logging, see [“Requirements, limitations, and support for file audit logging”](#) on page 489 and [“Installation prerequisites”](#) on page 352.

Note: A file system must be specified in the cluster definition file before you can enable file audit logging. You can configure the file audit logging related options in the cluster definition file by using the installation toolkit as follows.

By default, file audit logging is disabled in the cluster definition file.

- To enable file audit logging in the cluster definition file, issue the following command before doing installation or deployment with the installation toolkit:

```
./spectrumscale fileauditlogging enable
```

- To disable file audit logging in the cluster definition file, issue the following command:

```
./spectrumscale fileauditlogging disable
```

- To list the file audit logging configuration in the cluster definition file, issue the following command:

```
./spectrumscale fileauditlogging list
```

You can verify whether file audit logging is enabled in the cluster definition file by viewing the output of the **./spectrumscale node list** command:

```
[ INFO ] List of nodes in current configuration:
[ INFO ] [Installer Node]
[ INFO ] 198.51.100.15
[ INFO ] Setup Type: SpectrumScale
[ INFO ]
[ INFO ] [Cluster Name]
[ INFO ] ESDev1
[ INFO ]
[ INFO ] [Protocols]
[ INFO ] Object : Disabled
[ INFO ] SMB : Enabled
[ INFO ] NFS : Enabled
[ INFO ]
[ INFO ] File Audit logging : Disabled
[ INFO ]
[ INFO ] GPFS Node          Admin  Quorum  Manager  NSD Server  Protocol  GUI Server
OS Arch
[ INFO ] ESDev1-GPFS1        X      X      X              X
rhe17 x86_64
[ INFO ] ESDev1-GPFS2              X              X
rhe17 x86_64
[ INFO ] ESDev1-GPFS3              X              X
rhe17 x86_64
[ INFO ] ESDev1-GPFS4        X      X      X              X
rhe17 x86_64
[ INFO ] ESDev1-GPFS5        X      X      X              X
rhe17 x86_64
```

After enabling the file audit logging function in the cluster definition file, you must enable it on file systems on which you want to enable file audit logging.

- To enable file audit logging on a file system in the cluster definition file, issue the following command:

```
./spectrumscale filesystem modify --fileauditloggingenable FileSystemName
```

You can also specify the retention period and log fileset name with this command. For example, to specify a retention period of 180 days and to specify the log fileset name testlog, issue the following command:

```
./spectrumscale filesystem modify --fileauditloggingenable --retention 180 --logfileset testlog FileSystemName
```

- To disable file audit logging on a file system in the cluster definition file, issue the following command:

```
./spectrumscale filesystem modify --fileauditloggingdisable FileSystemName
```

Note: These file audit logging configuration-related changes become effective after the deployment procedure, initiated with **./spectrumscale deploy**, is completed.

Enabling and configuring call home using the installation toolkit

You can use the installation toolkit to enable and configure the call home function in the cluster definition file. Based on the call home configuration options specified in the cluster definition file using the installation toolkit, the call home function is enabled and configured in the cluster accordingly during the installation. By default, the call home function is enabled on all supported nodes in the cluster. If you plan to use the call home function, you must configure call home with the mandatory parameters by using the installation toolkit. For more information, see [this command example](#).

Note: If you want to only configure call home for a subset of the supported cluster nodes, or if you want to manually assign specific nodes to a specific call home group, see *Configuring call home to enable manual and automated data upload* in *IBM Spectrum Scale: Problem Determination Guide*.

The installation toolkit does not support all operating systems and platforms supported by call home. For example, the installation toolkit supports SLES and Ubuntu only on x86_64 and s390x, while call home also supports SLES and Ubuntu on Power.

The call home node needs to have network connectivity to the IBM Support Center because the call home node needs to upload data to IBM Support Center for all the members of its call home group. The call home node must be able to reach `https://esupport.ibm.com` either directly or using the proxy, which is specified in the call home configuration.

For more information about the call home function, see *Monitoring the IBM Spectrum Scale system by using call home* in *IBM Spectrum Scale: Problem Determination Guide*.

You can specify the call home node using the installation toolkit by issuing the following command:

```
./spectrumscale node add -c CallHomeNode
```

Note: This step is not mandatory. If a call home node is not specified, the installation toolkit assigns one of the nodes in the cluster as the call home node.

You can specify more than one node with this command. It is recommended to have at least one call home node for every 128 call home group members to prevent potential performance issues.

To configure call home options in the cluster definition file by using the installation toolkit, use the **./spectrumscale callhome** command as follows.

- To enable the call home in the cluster definition file by using the installation toolkit, issue the following command.

```
./spectrumscale callhome enable
```

A sample output is as follows.

```
[ INFO ] Enabling the callhome.
[ INFO ] Configuration updated.
```

Note: By default, call home is listed as enabled in the cluster definition file.

- To disable the call home in the cluster definition file by using the installation toolkit, issue the following command.

```
./spectrumscale callhome disable
```

A sample output is as follows.


```
[ INFO ] Disabling the callhome.  
[ INFO ] Configuration updated.
```

The call home function is enabled by default in the cluster definition file. If you disable it in the cluster definition file, when you issue the **./spectrumscale install** command, then no call home configuration is performed.

- To specify the call home configuration settings in the cluster definition file by using the installation toolkit, issue the **./spectrumscale callhome config** command. The following command example shows configuring the mandatory call home parameters:

```
./spectrumscale callhome config -n username -i 456123 -e username@example.com -cn US
```

A sample output is as follows.

```
[ INFO ] Setting Customer name to username  
[ INFO ] Setting Customer id to 456123  
[ INFO ] Setting Customer email to username@example.com  
[ INFO ] Setting Customer country to US
```

You are then prompted to accept or decline the support information collection message.

```
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
```

If you accept, the following output is displayed.

```
[ INFO ] License is accepted so the call home will be configured, if it is enabled.  
[ INFO ] Configuration is updated.
```

Note: You can specify the **-a** parameter with **./spectrumscale callhome config** to accept the support information collection message in advance.

- To clear the call home configuration specified in the cluster definition file by using the installation toolkit, issue the **./spectrumscale callhome clear** command. For example:

```
./spectrumscale callhome clear -i -e -cn
```

A sample output is as follows.

```
[ INFO ] Clearing Customer id  
[ INFO ] Clearing Customer email  
[ INFO ] Clearing Customer country
```

- To schedule the call home data collection in the cluster definition file by using the installation toolkit, issue the **./spectrumscale callhome schedule** command. For example:

```
./spectrumscale callhome schedule -d
```

A sample output is as follows.

```
[ INFO ] Setting scheduling to daily.  
[ INFO ] Configuration is updated.
```

If call home data collection is scheduled daily, data uploads are by default executed at 02:xx AM each day. If call home data collection is scheduled weekly, data uploads are by default executed at 03:xx AM each Sunday. In both cases, xx is a random number from 00 to 59.

You can use the **./spectrumscale callhome schedule -c** command to clear the call home data collection schedule.

- To list the call home configuration settings in the cluster definition file by using the installation toolkit, issue the following command.

```
./spectrumscale callhome list
```

A sample output is as follows.

```
[ INFO ] Current settings are as follows:
[ INFO ] Callhome is enabled.
[ INFO ] Schedule is daily.
[ INFO ] Setting Customer name to username
[ INFO ] Setting Customer id to 456123
[ INFO ] Setting Customer email to username@example.com
[ INFO ] Setting Customer country to US
[ WARN ] No value for Proxy ip in clusterdefinition file.
[ WARN ] No value for Proxy port in clusterdefinition file.
[ WARN ] No value for Proxy user in clusterdefinition file.
[ WARN ] No value for Proxy password in clusterdefinition file.
```

Note: These call home configuration-related changes become effective after the installation procedure, initiated with **./spectrumscale install**, is completed.

For detailed information on the **./spectrumscale callhome** command options, see **spectrumscale command** in *IBM Spectrum Scale: Command and Programming Reference*.

Installing IBM Spectrum Scale management GUI by using the installation toolkit

Use the following information for installing the IBM Spectrum Scale management GUI using the installation toolkit.

The IBM Spectrum Scale is only installed when nodes have been specified as GUI servers in the cluster definition with the **-g** option:

```
./spectrumscale node add gpfsnode3 -g
```

The IBM Spectrum Scale GUI requires passwordless ssh to all other nodes in the cluster. Therefore, the GUI server node must be added as an admin node using the **-a** flag:

```
./spectrumscale node add gpfsnode3 -a
```

If no nodes have been specified as GUI servers, then the GUI will not be installed. It is recommended to have 2 GUI interface servers and a maximum of 3 for redundancy.

The GUI will be installed on specified GUI servers when you run the **./spectrumscale install** command and on protocol nodes also specified as GUI servers during the **./spectrumscale deploy** command.

At the end of a successful GPFS installation or protocol deployment, you can access the GUI through a web browser with the following node address: `https://<GUI server IP or hostname>`.

Note: 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.

Populating cluster definition file with current cluster state using the installation toolkit

You can use the installation toolkit to populate the cluster definition file with the current cluster state.

Before using this functionality, review its limitations. For more information, see [“Limitations of config populate option of the installation toolkit”](#) on page 427.

Note: If you are using the object protocol or proxy enabled call home, the config populate functionality does not extract any password from the current cluster configuration. You must enter the password when prompted.

In the following scenarios, you might need to update the cluster definition file with the current cluster state:

- A manually created cluster in which you want to use the installation toolkit
- A cluster created using the installation toolkit in which manual changes were done

Use the installation toolkit to populate the cluster definition file as follows:

1. Define the installer node by issuing the **./spectrumscale setup -s InstallNode** command.
2. Repopulate the cluster definition file with the current cluster state by issuing the **./spectrumscale config populate --node Node** command.

Note: This command creates the backup of the existing cluster definition file and it creates a new cluster definition file.

- You can pass **--skip nsd** flag with config populate command to skip the nsd and filesystem details.

```
./spectrumscale config populate -N node --skip nsd
```

- In a cluster containing ESS, you must specify the EMS node with the config populate command. For example:

```
./spectrumscale config populate --node EMSNode
```

- You can also use this functionality to populate the cluster definition file with file audit logging and call home related information.

If you are performing an upgrade, proceed with the rest of the upgrade steps. For more information, see [“Upgrading IBM Spectrum Scale components with the installation toolkit” on page 543.](#)

Limitations of config populate option of the installation toolkit

You can use the **config populate** option of the installation toolkit to populate the cluster definition file with the current cluster state. This functionality has the following limitations.

Table 42. Limitations of the config populate functionality	
Function	Description
CES Groups	The config populate functionality does not detect CES groups. This might cause issues with CES stretch clusters
Clusters larger than 64 nodes	The config populate functionality does not support more than 64 nodes. For more information, see “Limitations of the installation toolkit” on page 394.
Configuration parameters such as pagepool	The config populate functionality does not detect configuration parameters such as pagepool.
Custom GPFS profiles	The config populate functionality does not detect custom profiles. This might not be an issue in most scenarios except when you plan to use the config populate functionality to back up a cluster's configuration for a future new installation.
File placement optimizer (FPO)	The config populate functionality does not detect FPO NSD stanza information. However, the existing FPO NSD stanza information is not affected.

Table 42. Limitations of the config populate functionality (continued)

Function	Description
Greater than 3 GUI nodes	The config populate functionality adds in GUI nodes properly if the number of GUI nodes is less than or equal to 3. However, if there are more than 3 GUI nodes, the config populate functionality does not add them. This might leave you with GUI nodes that are not at the latest level of code that must be manually upgraded, and other unexpected side effects.
Greater than 3 performance monitoring collector nodes	The config populate functionality does not support more than 3 performance monitoring collector nodes. If GUI nodes are detected, they are also assigned as the performance monitoring collector nodes.
Highly-available write cache (HAWC)	The config populate functionality does not detect HAWC information. However, the existing HAWC information is not affected.
Heterogeneous cluster	The config populate functionality can be used in a cluster that contains nodes with mixed CPU architectures with some restrictions. The installation toolkit populates the information of nodes in the cluster that have the same CPU architecture as the installer node. Information for nodes of a different CPU architecture is not populated.
IBM Spectrum Protect for Space Management and IBM Spectrum Archive Enterprise Edition (EE)	<p>The config populate functionality does not detect IBM Spectrum Protect for Space Management and IBM Spectrum Archive Enterprise Edition (EE) information.</p> <p>The installation toolkit provides some DMAPI detection and gives recommendations on how to proceed in case issues are encountered. For example: DMAPI holds an FS open for one of these functions, thereby causing upgrades to fail. In this case, the installation toolkit shows a hint that identifies the node and that it might need to be rebooted to proceed with the upgrade.</p>
IBM Spectrum Scale release 4.2.0.2	<p>In the IBM Spectrum Scale release 4.2.0.2, although the config populate functionality can be used to populate the cluster definition file with the current cluster state, not all parameters are properly updated. For example, CES IPs are not properly detected and you might need to add them manually by using the following command: ./spectrumscale config protocols -e ces_ip1,ces_ip2,...</p> <p>Review your cluster configuration after running ./spectrumscale config populate to ensure that all desired values are populated.</p>

Table 42. Limitations of the config populate functionality (continued)	
Function	Description
Local read-only cache (LROC)	The config populate functionality does not detect LROC information. However, the existing LROC information is not affected.
Non-federated performance monitoring setup	The config populate functionality does not support non-federated performance monitoring setup. The installation toolkit converts all performance monitoring configuration to federated mode unless specified to ignore performance monitoring completely.
NSD SAN attachment	The config populate functionality does not support retrieval of the NSD information in case of an NSD SAN attachment.
Offline mode in upgrade configuration	The config populate functionality cannot be used if one or more nodes in the cluster are designated as offline in the upgrade configuration.
Transparent cloud tiering	The config populate functionality does not detect Transparent cloud tiering related information.
Windows, AIX	<p>The config populate functionality does not support these operating systems.</p> <p>If any of these operating systems are detected on a node and if that node has the same CPU architecture and endianness as the node from which the installation toolkit is running, the config populate functionality shows a warning, skips the node, and continues populating information from other supported nodes.</p>

Related concepts

[“Limitations of the installation toolkit” on page 394](#)

Before using the installation toolkit to install IBM Spectrum Scale and deploy protocols, review the following limitations and workarounds, if any.

ESS awareness with the installation toolkit

You can use the installation toolkit to install IBM Spectrum Scale and deploy protocols in a cluster containing Elastic Storage Server or IBM Elastic Storage System (ESS).

When using the installation toolkit in a cluster containing ESS, use the following high-level steps:

1. Add protocol nodes in the ESS cluster by issuing the following command.

```
./spectrumscale node add NodeName -p
```

You can add other types of nodes such as client nodes, NSD servers, and so on depending on your requirements. For more information, see [“Defining the cluster topology for the installation toolkit” on page 409](#).

2. Specify one of the newly added protocol nodes as the installer node and specify the setup type by issuing the following command.

```
./spectrumscale setup -s NodeIP -i SSHIdentity -st ess
```

The installer node is the node on which the installation toolkit is extracted and from where the installation toolkit command, **spectrumscale**, is initiated.

3. Specify the EMS node of the ESS system to the installation toolkit by issuing the following command.

```
./spectrumscale node add NodeName -e
```

This node is also automatically specified as the admin node. The admin node, which must be the EMS node in an ESS configuration, is the node that has access to all other nodes to perform configuration during the installation.

You can use the `config populate` option of the installation toolkit to populate the cluster definition file with the current configuration of the EMS node, and the protocol and the client nodes in the cluster. In a cluster containing ESS, you must specify the EMS node with the `config populate` command. For example:

```
./spectrumscale config populate --node EMSNode
```

For more information, see [“Populating cluster definition file with current cluster state using the installation toolkit” on page 426.](#)

4. Proceed with specifying other configuration options, installing, and deploying by using the installation toolkit. For more information, see [“Defining the cluster topology for the installation toolkit” on page 409](#), [“Installing IBM Spectrum Scale and creating a cluster” on page 414](#), and [“Deploying protocols” on page 416.](#)

Several validations are done when you use the installation toolkit in an ESS cluster to ensure that functions that are configured on EMS and I/O server nodes are not affected. Additionally, some considerations apply when you are using the installation toolkit in an ESS cluster. These validations and considerations are as follows.

- Adding EMS node and I/O server nodes as protocol nodes is not allowed.
- Adding GUI nodes in the ESS cluster by using the installation toolkit is not allowed. It is assumed that the EMS node is running a GUI setup and the ability to add more GUI nodes by using the installation toolkit is disabled to not affect the existing configuration. During upgrade, the installation toolkit does not alter an existing configuration that contains one or more GUI setups. The upgrade of any of these components is done outside of the ESS EMS and I/O server nodes, and their configuration is not changed.
- Adding I/O server nodes as NSD nodes by using the installation toolkit is not allowed.
- Creating NSDs and file systems from disks within the ESS I/O server nodes by using the installation toolkit is not allowed.
- Specifying a cluster name by using the installation toolkit is not allowed. The existing cluster name is detected and used by the toolkit.
- CES shared root file system must be mounted on all protocol nodes.
- Only performance monitoring sensor packages are installed on all nodes in an ESS cluster other than the ESS EMS and I/O server nodes. They are configured to point to the existing collector on the EMS node. Apart from package installation, the installation toolkit adds protocol-specific sensors to the overall cluster performance monitoring configuration during the installation and deployment phases. During upgrade, the installation toolkit does not alter an existing configuration that might have one or more collectors or sensors. The upgrade of any of these components is done outside of the ESS EMS and I/O server nodes, and their configuration is not changed.

Related concepts

[“Using the installation toolkit to perform installation tasks: Explanations and examples” on page 407](#)

Use these explanations and examples of installation toolkit options and tasks to perform installation and deployment by using the installation toolkit.

Configuration of an IBM Spectrum Scale stretch cluster in an export services environment: a sample use case

This page describes a stretch cluster with NFS, SMB, and Object protocols that are enabled, installed, and deployed using the installation toolkit.

Note: File audit logging and clustered watch folder are not supported in a stretch cluster environment.

Overview of the stretch cluster use case

A single GPFS cluster is defined over three geographically separate sites: two production sites and a tiebreaker site. One or more file systems are created, mounted, and accessed concurrently from the two active production sites that are connected over a reliable WAN network.

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.

With two copies of the data in separate locations, if one site has an unrecoverable disaster, you can recover from a single site with no data loss. Data from two separate sites can share namespace and be accessed by either site. CES groups are enabled to control traffic to the local site. For more information, see *Synchronous mirroring with GPFS replication* in *IBM Spectrum Scale: Administration Guide*.

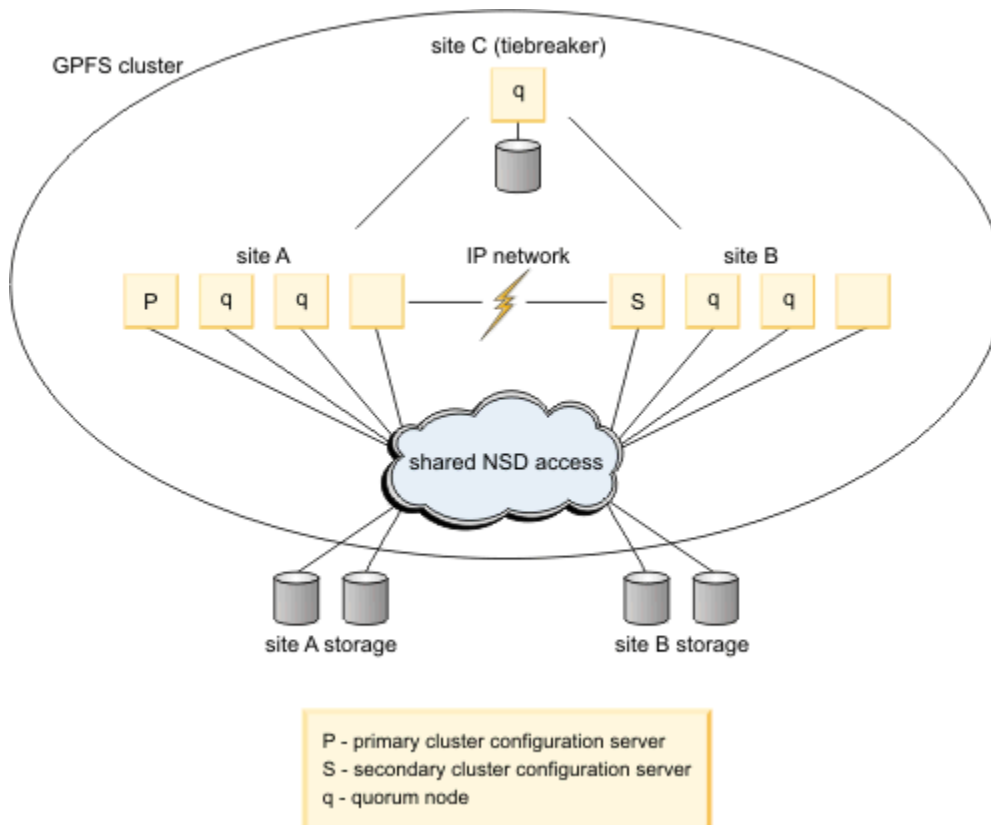


Figure 40. Synchronous mirroring with GPFS replication

About the tested use case

The stretch cluster in this use case was configured as active-active, meaning that clients can read and write data from either site of the cluster. For more information, see *Synchronous mirroring with GPFS replication* in *IBM Spectrum Scale: Administration Guide*. You can also configure an active-passive stretch cluster. For more information, see *An active-passive cluster* in *IBM Spectrum Scale: Administration Guide*. You can replicate data with GPFS in two ways: synchronous replication and asynchronous replication with Active File Management (AFM). Because there are some differences between the two options, you need to understand both options in order to choose the solution that best fits your use case.

Synchronous replication

Synchronous replication ensures that:

- your data is always available
- you can read and write in both locations
- you do not have to perform recovery actions from applications except for changing the IP address/hostname.

Because data is synchronously replicated, the application gets consistent data and no data is lost during failover or failback. Synchronous replication requires a reliable high-speed, low latency network between sites, and it has a performance impact on the application when the failback occurs.

Asynchronous replication with AFM

This use case covers the synchronous replication solution only. The use case shows how in an active-active stretch cluster when one site's storage is offline, you can still read from and write to a replica of the data to the site that is online.

Note: Stretch CES clusters with SMB must have low latencies. High latencies might result in performance degradation.

Asynchronous replication with AFM can work on a high latency network. Because data is asynchronously replicated, all updates might not be replicated when a failure occurs. Therefore, the application needs to be able to tolerate data loss and to run recovery operations to fail back. This setup requires two separate GPFS clusters instead of one cluster at multiple geographical sites.

Limitations of a stretch cluster that uses GPFS synchronous replication

Consider the following limitations of a stretch cluster with NFS, SMB, or Object for synchronously replicating data:

1. The IBM Spectrum Scale installation toolkit cannot deploy protocols if the CES networks across the two sites cannot communicate. For more information, see *Limitations* in *IBM Spectrum Scale: Big Data and Analytics Guide*.
2. If the Object protocol and the CES networks are separate and cannot communicate across the two sites, then object can use only one site to read and write data. For guidance on setup, refer to *Configuration of object for isolated node and network groups* in *IBM Spectrum Scale: Administration Guide*.
3. If your implementation requires you to set up IBM Spectrum Scale for object to one site, you will not have a seamless failover if you lose all of the protocol nodes on that site. You need to change the object ring configuration so that it points back to the CES Group that is available on the other site. For details, see *Configuration of object for isolated node and network groups* in *IBM Spectrum Scale: Administration Guide*.
4. When you have object enabled on one site and that entire site goes down unexpectedly, you might have to recover your endpoints manually since you can no longer ping them. In this case, refer to the steps provided in the OpenStack documentation: <https://docs.openstack.org/keystone/pike/install/keystone-install-rdo.html>.

Note: A deployment has a high chance of failure if the CES networks at each site cannot communicate with each other. For more information, see [“Limitations of the installation toolkit” on page 394](#). For

this use case, the cluster was deployed with protocols with separate VLAN'd networks on each site; however, those networks are able to communicate with each other.

Using the spectrumscale installation toolkit to install a stretch cluster

When you set up a stretch cluster, it is important to understand the physical setup of the storage and how it maps from each site to each file system and failure group. Figure 2 shows the tested configuration where each site (A and B) has storage that is only seen by the NSD servers in that site.

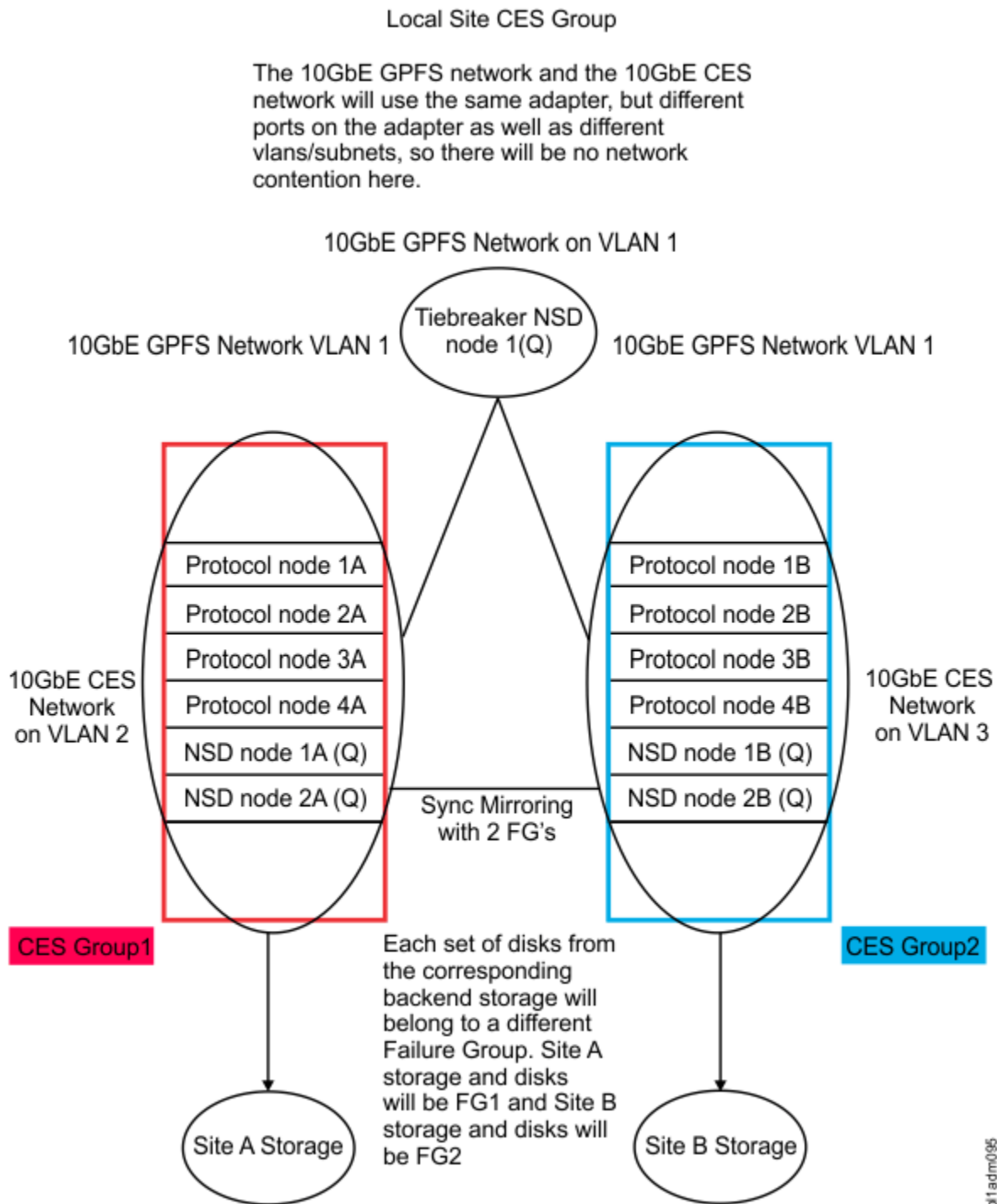


Figure 41. Local Site CES Group

For this use case example, the installation toolkit was used to install the IBM Spectrum Scale software. You can find the installation toolkit by changing directories to where it was extracted (the default 5.1.x.x extraction path follows. This path might vary depending on the code level).

```
cd /usr/lpp/mmfs/5.1.x.x/installer
```

Use these instructions to install and deploy a stretch cluster.

1. Designate a setup node by issuing the following command:

```
./spectrumscale setup -s InstallNodeIP
```

The setup node is used to run all of the toolkit commands and to specify the protocol and NSD nodes.

2. Specify the protocol and NSD nodes by issuing the following commands:

```
./spectrumscale node add protocol1A -a -p -g
./spectrumscale node add protocol2A -a -p -g
./spectrumscale node add protocol3A -p
./spectrumscale node add protocol4A -p
./spectrumscale node add protocol1B -p
./spectrumscale node add protocol2B -p
./spectrumscale node add protocol3B -p
./spectrumscale node add protocol4B -p
./spectrumscale node add nsd1A -n -q
./spectrumscale node add nsd2A -n -q
./spectrumscale node add nsd1B -n -q
./spectrumscale node add nsd2B -n -q
./spectrumscale node add nsd3C -n -q
```

The -s argument identifies the IP address that nodes use to retrieve their configuration. This IP address is one associated with a device on the installation node. (The IP address is automatically validated during the setup phase.)

The -q argument indicates the quorum nodes that are to be configured in the cluster. To keep the cluster accessible during a failure, configure most of the quorum nodes to have GPFS active. In this use case, there are five quorum nodes, therefore three must be active to keep the cluster accessible. These nodes were chosen specifically because they are the least likely to become inaccessible at the same time. Because nsd1A and nsd2A are at one site, nsd1B and nsd2B are at a second site, and nsd3C is at a third site, the likelihood of more than three going down at a time is minimal.

No manager nodes were specified with the -m argument, but by default, if no -m argument is specified, the installation toolkit automatically sets the protocol nodes to manager nodes, leaving an even balance across both sites.

The GUI node designations are specified with the -g argument to be on protocol nodes that reside on the same site, but you can choose to have a single GUI, two GUIs on one site, or two GUIs on different sites. In this case, two GUIs were tested on a single site.

3. Define NSD mappings to physical disks and assign those NSDs to failure groups and file systems. The following example NSDs are designated as dataAndMetadata; however, if you have the capacity (disk space and disk speed), set up Metadata disks on SSDs for the best performance.

```
./spectrumscale nsd add -p nsd1A -s nsd2A -u dataAndMetadata -fs ces -fg 2 /dev/mapper/lun_8
./spectrumscale nsd add -p nsd1B -s nsd2B -u dataAndMetadata -fs ces -fg 1 /dev/mapper/
lun_1

./spectrumscale nsd add -p nsd1B -s nsd2B -u dataAndMetadata -fs gpfs0 -fg 2 /dev/mapper/
lun_6
./spectrumscale nsd add -p nsd2B -s nsd1B -u dataAndMetadata -fs gpfs0 -fg 2 /dev/mapper/
lun_4
./spectrumscale nsd add -p nsd1B -s nsd2B -u dataAndMetadata -fs gpfs0 -fg 2 /dev/mapper/
lun_10
./spectrumscale nsd add -p nsd2B -s nsd1B -u dataAndMetadata -fs gpfs0 -fg 2 /dev/mapper/
lun_24
./spectrumscale nsd add -p nsd2A -s nsd1A -u dataAndMetadata -fs gpfs0 -fg 1 /dev/mapper/
lun_2
./spectrumscale nsd add -p nsd1A -s nsd2A -u dataAndMetadata -fs gpfs0 -fg 1 /dev/mapper/
lun_3
./spectrumscale nsd add -p nsd2A -s nsd1A -u dataAndMetadata -fs gpfs0 -fg 1 /dev/mapper/
lun_4
```

```
./spectrumscale nsd add -p nsd1A -s nsd2A -u dataAndMetadata -fs gpfs0 -fg 1 /dev/mapper/
lun_5

./spectrumscale nsd add -p nsd3C -u descOnly -fs gpfs0 -fg 3 /dev/sda
./spectrumscale nsd add -p nsd3C -u descOnly -fs ces -fg 3 /dev/sdb
```

Each file system, ces or gpfs0, has multiple disks that have primary and secondary servers at each site. This ensures that the file system stays online when an entire site goes down. With multiple primary and secondary servers for each disk and failure group that is local to each site, the GPFS replication keeps the data up to date across both sites. A disk with a primary and secondary server on site A belongs to failure group 1, and a disk with a primary and secondary server on site B belongs to failure group 2. This enables the two-way replication across the failure groups, meaning that one replica of data is kept at each site. The nsd3C node is known as the tiebreaker node. The physical disks that reside on that node /dev/sda and /dev/sdb are designated as ‘descOnly’ disks and are local to that node and are their own failure group. The descOnly argument indicates that the disk contains no file data or metadata. It is used solely to keep a copy of the file system descriptor. It is recommended to have that tiebreaker node in a separate geographical location than the other two sites.

4. Set up the file system characteristics for two-way replication on both the ces and gpfs0 file systems by issuing the following command:

```
./spectrumscale filesystem modify -r 2 -mr 2 ces
./spectrumscale filesystem modify -r 2 -mr 2 gpfs0
```

This sets the metadata and data replication to 2.

5. Designate file system paths for protocols and for object by issuing the following commands:

```
./spectrumscale config protocols -f ces -m /ibm/ces
./spectrumscale config object -f gpfs0 -m /ibm/gpfs0
```

6. Set the cluster name by issuing the following command:

```
./spectrumscale config gpfs -c gumby.tuc.stglabs.ibm.com
```

7. Install the stretch cluster by issuing the following command:

```
./spectrumscale install --precheck
./spectrumscale install
```

8. Set up the IP lists by issuing the following command:

```
./spectrumscale config protocols -e
10.18.52.30,10.18.52.31,10.18.52.32,10.18.52.33,10.18.60.30,10.18.60.31,10.18.60.32,10.18.60
.33
./spectrumscale filesystem list
```

9. Enable the protocols by issuing the following commands:

```
./spectrumscale enable nfs
./spectrumscale enable smb
./spectrumscale enable object
```

10. Configure object by issuing the following commands:

```
./spectrumscale config object -o Object_Fileset
./spectrumscale config object --adminpassword
./spectrumscale config object --databasepassword
```

11. Configure authentication by issuing the following command:

```
./spectrumscale auth file ad
./spectrumscale node list
```

12. Deploy the stretch cluster by issuing the following commands:

```
./spectrumscale deploy --precheck
./spectrumscale deploy
```

13. After the deployment completes, check the AD setup and status.

For the use case, the same AD server was on both sites, but you can use any authentication type in a stretch cluster that is supported on a single-site IBM Spectrum Scale cluster. Note that because a stretch cluster is still one cluster, more than one authentication method per site is not supported.

To check the status of the cluster's authentication issue either of these commands **mmuserauth service list** or **mmuserauth service check --server-reachability**.

Issue the **mmuserauth service list** command. The system displays information similar to the following:

```
FILE access configuration : AD
PARAMETERS                VALUES
-----
ENABLE_NFS_KERBEROS       false
SERVERS                   10.18.2.1
USER_NAME                 Administrator
NETBIOS_NAME              stretch_cluster
IDMAP_ROLE                master
IDMAP_RANGE               10000-1000000
IDMAP_RANGE_SIZE          10000
UNIXMAP_DOMAINS           DOMAIN1(10000000-29999999)
LDAPMAP_DOMAINS           none

OBJECT access configuration : LOCAL
PARAMETERS                VALUES
-----
ENABLE_KS_SSL             false
ENABLE_KS_CASIGNING       false
KS_ADMIN_USER             admin
```

Issue the **mmuserauth service check --server-reachability** command. The system displays information similar to the following:

```
Userauth file check on node: protocol1A
  Checking nsswitch file: OK
AD servers status
  NETLOGON connection: OK
  Domain join status: OK
  Machine password status: OK
Service 'gpfs-winbind' status: OK

Userauth object check on node: protocol1A
  Checking keystone.conf: OK
  Checking wsgi-keystone.conf: OK
  Checking /etc/keystone/ssl/certs/signing_cert.pem: OK
  Checking /etc/keystone/ssl/private/signing_key.pem: OK
  Checking /etc/keystone/ssl/certs/signing_cacert.pem: OK

Service 'httpd' status: OK
```

Possible steps to convert an IBM Spectrum Scale cluster to a stretch cluster

1. Add the nodes from the second and third sites to the original cluster either manually or by using the spectrumscale toolkit.
2. Create the tiebreaker disks on the third site.
3. If replicating an existing file system, use the **mmchfs** command to set the replicas of data and metadata blocks to 2. If you are creating a new file system, ensure that the replication factor is set to 2 when it is created. For details, see the section Using the spectrumscale installation toolkit to install a stretch cluster.
4. Restripe your file system by issuing the **mmrestripefs <filesystem> -R** command.
5. Enable CES on the protocol nodes that you have added to the configuration.
6. Create CES groups on both sites.

Configuring the stretch cluster

1. Set up some basic tuning parameters.

For the use case, the following tuning parameters were used to improve the performance and reliability of the cluster. Tuning parameters will vary significantly depending on the hardware resources in your environment.

```
mmchconfig readReplicaPolicy=fastest
mmchconfig unmountOnDiskFail=yes -N nsd3
mmchconfig workerThreads=1024 -N cesNode
mmchconfig -ipagepool=43G -N protocol1A
mmchconfig -ipagepool=31G -N protocol2A
mmchconfig pagepool=48G -N protocol3A
mmchconfig pagepool=48G -N protocol4A
mmchconfig pagepool=48G -N protocol1B
mmchconfig pagepool=48G -N protocol2B
mmchconfig pagepool=48G -N protocol3B
mmchconfig pagepool=48G -N protocol4B
mmchconfig pagepool=12G -N nsd1A
mmchconfig pagepool=16G -N nsd1B
mmchconfig pagepool=12G -N nsd2B
mmchconfig pagepool=12G -N nsd3C
mmchconfig maxFilesToCache=2M
mmchconfig maxMBPS=5000 -N cesNodes
```

For details on each parameter, see *Parameters for performance tuning and optimization in IBM Spectrum Scale: Administration Guide*. The use case was tested with `readReplicaPolicy=fastest` which is the recommended setting. A known limitation with `readReplicaPolicy=fastest` is that with networks that add ~3 ms latency (which are common in such installations) there is no substantial difference between local and remote disks (assuming the disk latency might be in the 40/50ms range). Thus, you might still read data from the remote site. Therefore, it is acceptable to use `readReplicaPolicy=local` to ensure the data is written/read on the local site as long as the local servers are on the same subnet as the clients and the remote servers are not. An NSD server on the same subnet as the client is also considered as local.

The **`readReplicaPolicy=fastest`** setting will work with either network topology, both sites on the same subnet or each site on its own subnet, as long as there is a measurable difference in the I/O access time.

2. Set up the CES nodes.

CES groups are needed when the CES networks on each site cannot communicate with each other. By having each site's local nodes in the same CES group, the administrator is able to control where the CES IPs failover to when there is an issue with a specific protocol node. If CES groups are not set up, a CES IP from Site A might attempt to failover to a node on Site B, and because there is no adapter for that IP to alias to on Site B (assuming different subnets), the failover will not succeed. CES groups make it easy to manage what CES nodes can host what CES IPs.

Set the CES nodes in the cluster to the corresponding groups by issuing the **`mmchnode --ces-group`** command (CES group names are not case-sensitive). For example:

```
mmchnode --ces-group SiteA -N protocol1A
mmchnode --ces-group SiteA -N protocol2A
mmchnode --ces-group SiteB -N protocol1B
mmchnode --ces-group SiteB -N protocol2B
```

In the example, protocol nodes `protocol1A` and `protocol2A` are set to the Site A CES group, protocol nodes `protocol1B` and `protocol2B` are set to the Site B CES group.

For detailed instructions, see *Setting up Cluster Export Services groups in an IBM Spectrum Scale cluster* in *IBM Spectrum Scale: Administration Guide*.

- Assign CES IPs to the corresponding CES groups. This ensures that IPs that reside on nodes in Site A do not fail over to nodes that reside in Site B and vice versa. Issue the **mmces address change command**. For example:

```
mmces address change --ces-ip
10.18.52.30,10.18.52.31,10.18.52.32,10.18.52.33 --ces-group SiteA
mmces address change --ces-ip
10.18.60.30,10.18.60.31,10.18.60.32,10.18.60.33 --ces-group SiteB
```

- To verify the CES groups your nodes belong to, issue the **mmces node list** command. The sample output is as follows:

Node	Name	Node Flags	Node Groups
10	protocol1B	none	siteB
11	protocol2B	none	siteB
12	protocol3B	none	siteB
13	protocol4B	none	siteB
6	protocol1A	none	siteA
7	protocol2A	none	siteA
8	protocol3A	none	siteA
9	protocol4A	none	siteA

- To verify the CES groups your CES IPs belong to, issue the **mmces address list** command. The sample output is as follows:

Address	Node	Group	Attribute
10.18.52.30	protocol1AsiteA	object_singleton_node	object_database_node
10.18.52.31	protocol2AsiteA	none	
10.18.52.32	protocol3AsiteA	none	
10.18.52.33	protocol4AsiteA	none	
10.18.60.30	protocol1BsiteB	none	
10.18.60.31	protocol2BsiteB	none	
10.18.60.32	protocol3BsiteB	none	
10.18.60.33	protocol4BsiteB	none	

A load balancer is recommended for the protocol stack to cater to a site loss. The load balancer will ensure that you do not encounter issues if using DNS Round Robin when a site goes down and that the host name in the DNS server can resolve all of the IP addresses.

Using NFS, SMB, and object with a stretch cluster

Using NFS and SMB protocols is similar to using a Spectrum Scale cluster that is in one geographical location. All clients can read and write to either site and to any CES IP that they connect with depending on access. If a single protocol node fails at one site, a normal IP failover will still occur within the site, and the client seamlessly fails over with NFS I/O continuing. SMB clients, however, might need to be reconnected. On failures, clients can reconnect to another cluster node because the IP addresses of failing nodes are transferred to another healthy cluster node. Windows SMB clients automatically open a new connection without additional intervention, but the application that is running I/O may need to be restarted. Object has a few more limitations. See the section *Limitations regarding a Stretch Cluster using GPFS synchronous replication* for details. In summary, if your CES networks cannot communicate across sites, you must choose a single site and its CES group to configure with object. During a full site outage, you will need to make the manual fixes described in the Limitations section. A single protocol node failure will still work but you will need to retry after the CES IP moves to a new node within the CES group.

Monitoring and administering a stretch cluster

Monitoring your stretch cluster is the same as monitoring a cluster in a single location, except for the disk setup, and knowing when your disks are down and what site is affected. You can see the disk status using the **mmlsdisk** command. The sample output is as follows:

disk name	driver type	sector size	failure group	holds metadata	holds data	status	storage availability	pool
nsd22	nsd	512	2	Yes	Yes	ready	up	system
nsd23	nsd	512	2	Yes	Yes	ready	up	system
nsd24	nsd	512	2	Yes	Yes	ready	up	system
nsd25	nsd	512	2	Yes	Yes	ready	up	system
nsd26	nsd	512	1	Yes	Yes	ready	up	system
nsd27	nsd	512	1	Yes	Yes	ready	up	system
nsd28	nsd	512	1	Yes	Yes	ready	up	system
nsd29	nsd	512	1	Yes	Yes	ready	up	system
nsd30	nsd	512	1	Yes	Yes	ready	up	system
nsd31	nsd	512	3	No	No	ready	up	system

A healthy cluster shows all the disks that have a status of up. You can also verify the replication settings using the **mmlsfs -m -M -r -R** command. The sample output is as follows:

flag	value	description
-m	2	Default number of metadata replicas
-M	3	Maximum number of metadata replicas
-r	2	Default number of data replicas
-R	3	Maximum number of data replicas

If the default number of data and metadata replicas is set to two, this will indicate that you have no disk failures and that your data is being replicated across both failure groups.

Issue the **mmlsdisk ces** command. The sample output is as follows:

disk name	driver type	sector size	failure group	holds metadata	holds data	status	storage availability	pool
nsd20	nsd	512	2	Yes	Yes	ready	up	system
nsd21	nsd	512	1	Yes	Yes	ready	up	system
nsd32	nsd	512	3	No	No	ready	up	system

If you lose access to one site's storage due to maintenance, network issues, or hardware issues, the disks in the cluster are marked as down and the **mmhealth node show** command results shows them as down. This is acceptable because the stretch cluster can keep operating when an entire site goes down. There can be a negative impact on performance while one site is down, but that is expected.

To see the disk cluster status for the use case, issuing the **mmlsdisk gpfs0** command shows the following information:

disk name	driver type	sector size	failure group	holds metadata	holds data	status	storage availability	pool
nsd22	nsd	512	2	Yes	Yes	ready	down	system
nsd23	nsd	512	2	Yes	Yes	ready	down	system
nsd24	nsd	512	2	Yes	Yes	ready	down	system
nsd25	nsd	512	2	Yes	Yes	ready	down	system
nsd26	nsd	512	1	Yes	Yes	ready	up	system
nsd27	nsd	512	1	Yes	Yes	ready	up	system
nsd28	nsd	512	1	Yes	Yes	ready	up	system
nsd29	nsd	512	1	Yes	Yes	ready	up	system
nsd30	nsd	512	1	Yes	Yes	ready	up	system
nsd31	nsd	512	3	No	No	ready	up	system

For the use case, the results of the **mmhealth node show -N nsd2B disk** command show three disks:

```
Node name:      nsd2B
Node status:    FAILED
Status Change:  17 min. ago
Component      Status      Status Change  Reasons
-----
GPFS           FAILED      17 min. ago    gpfs_down, quorum_down
NETWORK        HEALTHY     10 days ago    -
FILESYSTEM     DEPEND     17 min. ago    unmounted_fs_check(gpfs1, ces, gpfs0)
DISK           DEPEND     17 min. ago    disk_down(nsd20, nsd22, nsd23)
PERFMON        HEALTHY     10 days ago    -
```

To see all of the failed disks, issue the **mmhealth node show nsd2B** command (without the -N attribute). For the use case, the system displays the following information:

```
Node name:      nsd2B
Component      Status      Status Change  Reasons
-----
DISK           DEPEND     18 min. ago    disk_down(nsd20, nsd22, nsd23)
nsd1           DEPEND     18 min. ago    -
nsd10          DEPEND     18 min. ago    -
nsd11          DEPEND     18 min. ago    -
nsd12          DEPEND     18 min. ago    -
nsd13          DEPEND     18 min. ago    -
nsd14          DEPEND     18 min. ago    -
nsd15          DEPEND     18 min. ago    -
nsd16          DEPEND     18 min. ago    -
nsd17          DEPEND     18 min. ago    -
nsd18          DEPEND     18 min. ago    -
nsd19          DEPEND     18 min. ago    -
nsd2           DEPEND     18 min. ago    -
nsd20          DEPEND     18 min. ago    disk_down
nsd22          DEPEND     18 min. ago    disk_down
nsd23          DEPEND     18 min. ago    disk_down
nsd24          DEPEND     18 min. ago    disk_down
nsd25          DEPEND     18 min. ago    disk_down
nsd3           DEPEND     18 min. ago    -
nsd4           DEPEND     18 min. ago    -
nsd5           DEPEND     18 min. ago    -
nsd6           DEPEND     18 min. ago    -
nsd7           DEPEND     18 min. ago    -
nsd8           DEPEND     18 min. ago    -
nsd9           DEPEND     18 min. ago    -

Event          Parameter    Severity      Active Since  Event Message
-----
disk_down nsd20      WARNING       16 min. ago   Disk nsd20 is reported as not up
disk_down nsd22      WARNING       16 min. ago   Disk nsd22 is reported as not up
disk_down nsd23      WARNING       16 min. ago   Disk nsd23 is reported as not up
disk_down nsd24      WARNING       16 min. ago   Disk nsd24 is reported as not up
disk_down nsd25      WARNING       16 min. ago   Disk nsd25 is reported as not up
```

After the issue is resolved, restart the disks and make sure that the data and metadata replicas are intact. First, ensure that GPFS is active on all nodes. Next, issue the **mmchdisk <filesystem> start-a** command. This informs GPFS to try to access the disks that are marked down and, if possible, to move them back into the up state. This is accomplished 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 changed to unrecovered. This could occur if too many disks are down. The metadata scan can be re-initiated later by issuing the **mmchdisk** command again. If more than one disk in the file system is down, all of the disks that are down must be started at the same time by issuing **mmchdisk <filesystem> start -a**. If you start them separately and metadata is stored on any disk that remains down, the **mmchdisk start** command fails.

```
mmnsddiscover: Attempting to rediscover the disks. This may take a while ...
mmnsddiscover: Finished.
nsd2A: Rediscovered nsd server access to nsd26.
nsd2A: Rediscovered nsd server access to nsd28.
nsd3C: Rediscovered nsd server access to nsd31.
sdn2B: Rediscovered nsd server access to nsd23.
```



```

nsd1B: Rediscovered nsd server access to nsd23.
nsd2B: Rediscovered nsd server access to nsd24.
nsd1B: Rediscovered nsd server access to nsd24.
nsd1A: Rediscoverensd server access to nsd29.
nsd2A: Rediscovered nsd server access to nsd30.
nsd2A: Rediscovered red nsd server access to nsd27.
nsd2B: Rediscovered nsd server access to nsd25.
nsd2B: Rediscovered nsd server access to nsd22.
nsd2A: Rediscovered nsd server access to nsd25.
nsd2A: Rediscovered nsd server access to nsd22.
Scanning file system metadata, phase 1 ...
33 % complete on Fri Feb 3 11:46:41 2017
66 % complete on Fri Feb 3 11:56:57 2017
100 % complete on Fri Feb 3 11:58:24 2017 Scan completed successfully.
Scanning file system metadata, phase 2 ...
Scan completed successfully.
Scanning file system metadata, phase 3 ...
8 % complete on Fri Feb 3 11:58:29 2017
16 % complete on Fri Feb 3 11:58:32 2017
23 % complete on Fri Feb 3 11:58:35 2017
...
91 % complete on Fri Feb 3 11:59:18 2017
95 % complete on Fri Feb 3 11:59:22 2017
98 % complete on Fri Feb 3 11:59:25 2017
100 % complete on Fri Feb 3 11:59:26 2017
Scan completed successfully.
Scanning file system metadata, phase 4 ...
Scan completed successfully.
Scanning user file metadata ...
2.37 % complete on Fri Feb 3 11:59:46 2017 ( 2473984 inodes with total 672770 MB data
processed)
3.86 % complete on Fri Feb 3 12:00:07 2017 ( 4734976 inodes with total 1094807 MB data
processed)
4.59 % complete on Fri Feb 3 12:00:27 2017 ( 7880704 inodes with total 1301307 MB data
processed)
5.30 % complete on Fri Feb 3 12:00:47 2017 ( 11003904 inodes with total 1501577 MB data
processed)
6.01 % complete on Fri Feb 3 12:01:07 2017 ( 14077952 inodes with total 1703928 MB data
processed)
6.70 % complete on Fri Feb 3 12:01:27 2017 ( 17154048 inodes with total 1896877 MB data
processed)
7.36 % complete on Fri Feb 3 12:01:47 2017 ( 20135936 inodes with total 2084748 MB data
processed)
7.97 % complete on Fri Feb 3 12:02:07 2017 ( 22512640 inodes with total 2257626 MB data
processed)
8.21 % complete on Fri Feb 3 12:02:27 2017 ( 23322624 inodes with total 2327269 MB data
processed)
8.39 % complete on Fri Feb 3 12:02:48 2017 ( 24182784 inodes with total 2377108 MB data
processed)
8.52 % complete on Fri Feb 3 12:03:09 2017 ( 25182208 inodes with total 2414040 MB data
processed)
8.64 % complete on Fri Feb 3 12:03:29 2017 ( 26166272 inodes with total 2447380 MB data
processed)
...
96.58 % complete on Fri Feb 3 12:36:40 2017 ( 198458880 inodes with total 27362407 MB data
processed)
96.82 % complete on Fri Feb 3 12:37:00 2017 ( 202438144 inodes with total 27430464 MB data
processed)
97.06 % complete on Fri Feb 3 12:37:20 2017 ( 206526720 inodes with total 27498158 MB data
processed)
97.30 % complete on Fri Feb 3 12:37:40 2017 ( 210588672 inodes with total 27567944 MB data
processed)
97.46 % complete on Fri Feb 3 12:38:00 2017 ( 266730496 inodes with total 27612826 MB data
processed)
97.52 % complete on Fri Feb 3 12:38:20 2017 ( 302344960 inodes with total 27629694 MB data
processed)
97.59 % complete on Fri Feb 3 12:38:40 2017 ( 330066432 inodes with total 27648547 MB data
processed)
100.00 % complete on Fri Feb 3 12:38:52 2017 ( 394185216 inodes with total 27657707 MB data
processed)
Scan completed successfully.

```

The recovery time for this command can vary depending on how much data was written while the disks were down. If the disks were down for a long time (greater than 24 hours) and a lot of data was written in that time, it is expected that the **mmchdisk** command could take quite a while to complete. The time needed to bring up the disks depends on the quantity of data changed during the time the disks were down. This command is run while the file data remains accessible to the applications so I/O clients can continue to operate.

IBM Spectrum Scale stretch cluster use case conclusion

Each use case for a stretch cluster will vary. The sample use case represents one tested configuration. For more information see the following topics:

Synchronous mirroring with GPFS replication in IBM Spectrum Scale: Administration Guide.

Synchronous mirroring with GPFS replication in IBM Spectrum Scale: Administration Guide.

An active-passive cluster in IBM Spectrum Scale: Administration Guide

Limitations in IBM Spectrum Scale: Big Data and Analytics Guide

Configuration of object for isolated node and network groups in IBM Spectrum Scale: Administration Guide

<https://docs.openstack.org/keystone/pike/install/keystone-install-rdo.html>

Parameters for performance tuning and optimization in IBM Spectrum Scale: Administration Guide

Setting up Cluster Export Services groups in an IBM Spectrum Scale cluster in IBM Spectrum Scale: Administration Guide

Performing additional tasks using the installation toolkit

Use this information for using the installation toolkit for tasks such as deploying protocols on existing clusters and adding nodes to an existing installation.

Deploying protocols on an existing cluster

Use this information to deploy protocols on an existing cluster by using the installation toolkit.

1. Designate the installer node by using the following command.

```
./spectrumscale setup -s Installer_Node_IP
```

2. Designate protocol nodes in your environment to use for the deployment by using the following command.

```
./spectrumscale node add FQDN -p
```

3. Designate GUI nodes in your environment to use for the deployment by using the following command.

```
./spectrumscale node add FQDN -g
```

4. Configure protocols to point to a shared root file system by using the following command.

```
./spectrumscale config protocols -f FS_Name -m FS_Mountpoint
```

5. Configure a pool of export IP addresses by using the following command.

```
./spectrumscale config protocols -e Comma_Separated_List_of_Exportpool_IPs
```

6. Enable NFS on all protocol nodes by using the following command.

```
./spectrumscale enable nfs
```

7. Enable SMB on all protocol nodes by using the following command.

```
./spectrumscale enable smb
```

8. Enable object on all protocol nodes by using the following command.

```
./spectrumscale enable object
./spectrumscale config object -au admin -ap -dp
./spectrumscale config object -e FQDN
./spectrumscale config object -f FS_Name -m FS_Mountpoint
./spectrumscale config object -o Object_Fileset
```

9. Review the configuration before deployment by using the following command.

```
./spectrumscale config protocols
./spectrumscale config object
./spectrumscale node list
```

10. Deploy protocols on your defined environment by using the following command.

```
./spectrumscale deploy --precheck
./spectrumscale deploy
```

Adding nodes, NSDs, or file systems to an existing cluster

Use this information to add nodes, NSDs, or file systems to an existing cluster by using the installation toolkit.

- To add nodes to an existing cluster, do the following.

a) Add one or more node types by using the following commands.

Note: For adding nodes to an existing cluster, ensure that the IBM Spectrum Scale version on the nodes that you are adding is the same as the version on the existing nodes in the cluster.

– Client nodes:

```
./spectrumscale node add FQDN
```

– NSD nodes:

```
./spectrumscale node add FQDN -n
```

– Protocol nodes:

```
./spectrumscale node add FQDN -p
```

– GUI nodes:

```
./spectrumscale node add FQDN -g -a
```

For the list of supported node types that can be added, see *spectrumscale command* in *IBM Spectrum Scale: Administration Guide*.

Important: The installation toolkit does not alter anything on the existing nodes in the cluster. You can determine the existing nodes in the cluster by using the **mm1sc1uster** command.

The installation toolkit might change the performance monitoring collector configuration if you are adding the new node as a GUI node or an NSD node, due to collector node prioritization. However, if you do not want to change the collector configuration then you can use the **./spectrumscale config perfmon -r off** command to disable performance monitoring before initiating the installation procedure.

b) Install IBM Spectrum Scale on the new nodes by using the following commands.

```
./spectrumscale install --precheck
./spectrumscale install
```

c) If protocol nodes are being added, deploy protocols by using the following commands.

```
./spectrumscale deploy --precheck
./spectrumscale deploy
```

- To add NSDs to an existing cluster, do the following.

a) Verify that the NSD server that connects this new disk runs an OS compatible with the installation toolkit and that the NSD server exists within the cluster.

b) Add NSDs to the cluster definition by using the following command.

```
./spectrumscale nsd add -p FQDN_of_Primary_NSD_Server Path_to_Disk_Device_File
```

c) Run the installation by using the following commands.

```
./spectrumscale install --precheck  
./spectrumscale install
```

- To add file systems to an existing installation, do the following.
 - a) Verify that free NSDs exist and that they can be listed by the installation toolkit by using the following commands.

```
mmlsnsd  
./spectrumscale nsd list
```

b) Define the file system by using the following command.

```
./spectrumscale nsd modify NSD -fs File_System_Name
```

c) Install the file system by using the following commands.

```
./spectrumscale install --precheck  
./spectrumscale install
```

Enabling another protocol on an existing cluster that has protocols enabled

Use this information to enable protocols on an existing cluster that has other protocols that are enabled by using the installation toolkit.

1. Use one of the following steps depending on your configuration.

- Enable NFS on all protocol nodes by using the following command.

```
./spectrumscale enable nfs
```

- Enable SMB on all protocol nodes by using the following command.

```
./spectrumscale enable smb
```

- Enable Object on all protocol nodes by using the following commands.

```
./spectrumscale enable object  
./spectrumscale config object -au Admin_User -ap Admin_Password -dp Database_Password  
./spectrumscale config object -e FQDN  
./spectrumscale config object -f FS_Name -m FS_Mountpoint  
./spectrumscale config object -o Object_Fileset
```

2. Enable the new protocol by using the following command.

```
./spectrumscale deploy --precheck  
./spectrumscale deploy
```

Diagnosing errors during installation, deployment, or upgrade

Use this information to diagnose errors during installation, deployment, or upgrade by using the installation toolkit.

1. Note the screen output that is indicating the error. This output helps in narrowing down the general failure.

When a failure occurs, the screen output also shows the log file that contains the failure message.

2. Open the log file in an editor such as **vi**.

3. Go to the end of the log file and search upwards for the text FATAL.

4. Find the topmost occurrence of FATAL (or first FATAL error that occurred) and look in preceding and following sections of this error message for further indications of the failure.

For more information, see *Finding deployment related error messages more easily and using them for failure analysis* in *IBM Spectrum Scale: Problem Determination Guide*.

Preparing a cluster that contains ESS for adding protocols

Use this information to prepare a cluster that contains ESS for adding functions such as NFS, SMB, object, GUI, and performance monitoring.

Before you add protocols to a cluster that contains ESS, ensure that:

- The cluster that contains ESS is active and online.
- Red Hat Enterprise Linux 7.x is installed on nodes that are going to serve as protocol nodes.
- Red Hat Enterprise Linux 7.x base repository is set up on nodes that are going to serve as protocol nodes.

1. Use the ESS GUI or CLI to create a CES shared root file system.

Note: The CES shared root file system must be at least 4 GB and it must not be encrypted.

2. On all nodes that are going to serve as protocol nodes, download and extract the IBM Spectrum Scale protocols standard or advanced packages.

For information on extracting packages, see [“Extracting the IBM Spectrum Scale software on Linux nodes”](#) on page 355.

3. On all nodes that are going to serve as protocol nodes, install core GPFS RPMs from the `/usr/lpp/mmfs/5.x.x.x/gpfs_rpms`.



Attention: Ensure that you do not install packages such as `perfmon`, `gui`, `callhome`, `java`, and `protocols` now. These components are installed and configured with the protocols deployment.

The core GPFS RPMs that need to be installed include:

- `gpfs.base`
 - `gpfs.gpl`
 - `gpfs.license.xx`
 - `gpfs.gskit`
 - `gpfs.docs`
 - `gpfs.msg.en_US`
 - `gpfs.compression`
 - `gpfs.adv` (optional)
 - `gpfs.crypto` (optional)
4. Add the nodes that are going to serve as protocol nodes to the cluster by using the **`mmaddnode`** command.
 5. Enable the Cluster Configuration Repository (CCR) on the cluster if it is not enabled already.



Attention: If GPFS levels between protocol nodes and ESS differ significantly, ensure that the nodes with the newer code level of GPFS are designated as both quorum and manager nodes. For example, old ESS systems with GPFS 4.1.0-8 are incompatible with CES. These ESS systems can be a part of a cluster with protocols only if they are not designated as quorum and manager nodes.

This aspect requires careful planning and administration because:

- By default, ESS nodes are designated as quorum and manager nodes.
 - In some cases, there might not be extra nodes outside of ESS to make sure that only nodes with the newer code level are designated as quorum and manager nodes.
6. Verify that passwordless SSH is working between all nodes in the cluster.
 7. Verify that firewall ports are set correctly.
 8. Verify that the CES shared root file system is mounted and that it is set to auto mount.
 9. Set NFSv4 ACLs for file systems that are going to be used for protocol data, if they are not set already.

10. On one of the nodes that are going to serve as protocol nodes, run the installation toolkit to start deploying protocols.

Note:

- Designate only the nodes that you plan to use as protocol nodes. Do not designate existing ESS nodes such as EMS or I/O nodes as protocol nodes.
- Point to existing file systems only. Do not attempt to create new file systems or NSDs for this purpose.

For information about deploying protocols in a cluster, see **spectrumscale command** in *IBM Spectrum Scale: Command and Programming Reference* and [“Deploying protocols on an existing cluster” on page 442](#).

Related concepts

[“ESS awareness with the installation toolkit” on page 429](#)

You can use the installation toolkit to install IBM Spectrum Scale and deploy protocols in a cluster containing Elastic Storage Server or IBM Elastic Storage System (ESS).

Adding an IBM Spectrum Archive Enterprise Edition (EE) node in an IBM Spectrum Scale cluster

Use this information to add an IBM Spectrum Archive Enterprise Edition (EE) node in an IBM Spectrum Scale cluster.

1. Use the installation toolkit to install IBM Spectrum Scale and create a cluster.

For more information, see [“Using the installation toolkit to perform installation tasks: Explanations and examples” on page 407](#).

2. If the node on which IBM Spectrum Archive Enterprise Edition (EE) is to be installed is running on RHEL 6.x, install IBM Spectrum Scale packages on that node manually by using the RPMs from a protocol node.
3. Add the IBM Spectrum Archive Enterprise Edition (EE) node to your cluster at any time after the cluster is created.

For more information, see *Adding nodes to a GPFS cluster* in *IBM Spectrum Scale: Administration Guide*.

4. Before you install IBM Spectrum Archive Enterprise Edition (EE), enable Data Management API (DMAPI) on the file system to be used with IBM Spectrum Archive Enterprise Edition (EE).

- a) Unmount the file system to be used with IBM Spectrum Archive Enterprise Edition (EE).

Note: If this file system is also used for protocols, you cannot unmount the file system unless you issue the **mmshutdown** command on all protocol nodes.

- b) Enable DMAPI by using the **mmchfs Device -z yes** command.

In this command example, *Device* is the device name of the file system.

- c) If you used the **mmshutdown** command in a preceding step, mount the file system again and issue the **mmstartup** on all protocol nodes.

5. Install and configure IBM Spectrum Archive Enterprise Edition (EE).

For detailed information on using IBM Spectrum Archive and IBM Spectrum Scale together, see [Active Archive Implementation Guide with Spectrum Scale Object and IBM Spectrum Archive Redpaper](#). Chapters 2 and 3 of this document contain specific details about cluster configuration and how to add the IBM Spectrum Archive node to the IBM Spectrum Scale cluster.

Protocol node IP further configuration

The **mmces** commands for adding or moving IPs are as follows.

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

Checking that protocol IPs are currently configured

The `mmces address list` command shows all the currently configured protocol IPs.

Note: The term CES IP refers to the Cluster Export Services IP, or a protocol IP used specifically for NFS, SMB, HDFS, and Object protocol access. All CES IPs must have an associated hostname and reverse DNS lookup must be configured for each. For more information, see *Adding export IPs* in [“Deploying protocols”](#) on page 416.

```
mmces address list
```

The system displays output similar to the following:

```
Address Node Group Attribute
-----
10.0.0.101 prt001st001 none none
10.0.0.102 prt002st001 none none
10.0.0.103 prt003st001 none none
10.0.0.104 prt004st001 none none
10.0.0.105 prt005st001 none none
10.0.0.106 prt006st001 none none
```

Additional IPs can be added to a node. In this example, IPs `10.0.0.108` and `10.0.0.109` are added to protocol node 5 (`prt005st001`).

```
mmces address add --ces-node prt005st001 --ces-ip 10.0.0.108,10.0.0.109
mmces address list
```

The system displays output similar to the following:

```
Address Node Group Attribute
-----
10.0.0.101 prt001st001 none none
10.0.0.102 prt002st001 none none
10.0.0.103 prt003st001 none none
10.0.0.104 prt004st001 none none
10.0.0.105 prt005st001 none none
10.0.0.106 prt006st001 none none
10.0.0.108 prt005st001 none none
10.0.0.109 prt005st001 none none
```

IPs can also be moved among nodes manually. This is helpful if IP allocation becomes unbalanced among the nodes. Continuing from the prior example, `prt005st001` now has three protocol IPs whereas all other nodes have a single protocol IP. To balance this out a bit better, `10.0.0.109` will be manually moved to node `prt004st001`.

```
mmces address move --ces-ip 10.0.0.109 --ces-node prt004st001
mmces address list
```

The system displays output similar to the following:

```
Address Node Group Attribute
-----
10.0.0.101 prt001st001 none none
10.0.0.102 prt002st001 none none
10.0.0.103 prt003st001 none none
10.0.0.104 prt004st001 none none
10.0.0.105 prt005st001 none none
10.0.0.106 prt006st001 none none
10.0.0.108 prt005st001 none none
10.0.0.109 prt004st001 none none
```

List all protocol services enabled

Run the following command from any protocol node to list the running services on all protocol nodes:

```
mmces service list -a
```

The system displays output similar to the following:

```
Enabled services: SMB NFS OBJ HDFS
prt001st001: SMB is not running, NFS is not running, OBJ is running, HDFS is running,
prt002st001: SMB is running, NFS is running, OBJ is not running, HDFS is running
prt003st001: SMB is running, NFS is running, OBJ is running, HDFS is running
```

Object protocol further configuration

If the Object protocol is enabled during installation, use the following information to verify the Object protocol configuration. You can also enable features such as unified file and object access and multi-region object deployment.

Object is set up, enabled, and configured by the installation toolkit, but a few extra steps are necessary to ready the Object protocol for use. Verify the Object protocol by running a few tests.

Verifying the Object protocol portion of the installation

Perform the following simple example OpenStack client and unified file and object access commands to verify your installation. You can list all users and projects (called accounts in unified file and object access), list the unified file and object access containers, upload a sample object to a container, and list that container and see the object. These commands complete with no errors.

```
# source $HOME/openrc
# openstack user list
# openstack project list
# swift stat
# date > object1.txt
# swift upload test_container object1.txt
object1.txt
# swift list test_container
object1.txt
```

When the SELinux mode is Enforcing and a user on the protocol node needs to run Swift helper commands, the **runcon** command must be used. Otherwise, the system might generate SELinux failure logs. For example, the *.recon files in the /var/cache/swift directory are updated by Swift helper services such as replicator and updater.

1. To run the helper command, type: `runcon -t swift_t -r system_r swift helper command`
2. To check whether the SELinux context is correct, run the **matchpathcon** command: `matchpathcon -V /var/cache/swift/*`

The system displays the following output:

```
/var/cache/swift/account.recon verified.
/var/cache/swift/container.recon has context
unconfined_u:object_r:var_t:s0, should be system_u:object_r:swift_var_cache_t:s0
/var/cache/swift/object.recon verified.
```

3. To fix the SELinux context in the file, run the **restorecon** command: `restorecon /var/cache/swift/container.recon`

For more information about SELinux consideration, see [“SELinux considerations” on page 323](#).

Enabling multi-region object deployment initially

For multi-region object deployment, each region is a separate cluster and on each cluster IBM Spectrum Scale for object storage is installed independently by using the installer.

In a single cluster, the installer completely installs the object protocol on all the nodes within that cluster. However, in a multi-region object deployment environment, each cluster is installed independently and the object protocol on the independent clusters is linked together.

To set up an initial multi-region environment, run the following command on the first cluster after it is installed:

```
mmobj multiregion enable
```

You can add a region to a multi-region deployment environment. For more information, see *Adding a region in a multi-region object deployment* in *IBM Spectrum Scale: Administration Guide*.

Related concepts

[Installing and using unified file and object access](#)

Unified file and object access are installed when you install IBM Spectrum Scale for object storage in an IBM Spectrum Scale cluster. No additional steps are required for installing unified file and object access.

Related tasks

[Enabling unified file and object access after upgrading from IBM Spectrum Scale 4.2 or later](#)

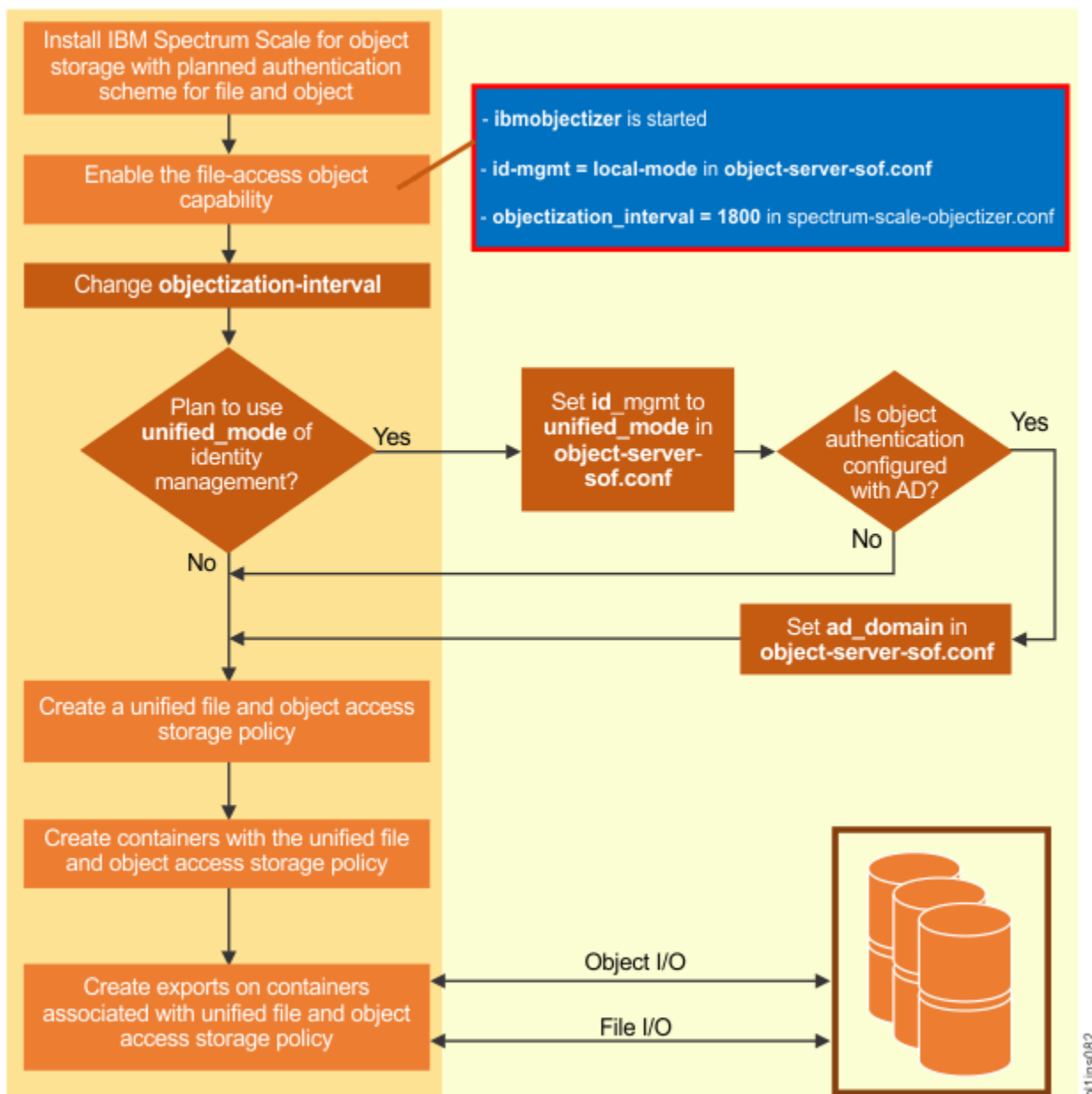
Use these steps to enable unified file and object access after you upgrade from IBM Spectrum Scale Version 4.2 or later.

Installing and using unified file and object access

Unified file and object access are installed when you install IBM Spectrum Scale for object storage in an IBM Spectrum Scale cluster. No additional steps are required for installing unified file and object access.

You can manage and administer unified and object access. For more information, see *Unified file and object access in IBM Spectrum Scale* in *IBM Spectrum Scale: Administration Guide*.

The high-level flow of administering unified file and object access is shown in the following diagram. The flow of changing the identity management mode for unified file and object access with the **id_mgmt** parameter is also shown.



For detailed steps, see *Administering unified file and object access* in *IBM Spectrum Scale: Administration Guide*.

Related concepts

Enabling multi-region object deployment initially

For multi-region object deployment, each region is a separate cluster and on each cluster IBM Spectrum Scale for object storage is installed independently by using the installer.

Related tasks

Enabling unified file and object access after upgrading from IBM Spectrum Scale 4.2 or later

Use these steps to enable unified file and object access after you upgrade from IBM Spectrum Scale Version 4.2 or later.

Enabling unified file and object access after upgrading from IBM Spectrum Scale 4.2 or later

Use these steps to enable unified file and object access after you upgrade from IBM Spectrum Scale Version 4.2 or later.

1. Enable the file-access object capability.

For detailed steps, see *Enabling the file-access object capability* in *IBM Spectrum Scale: Administration Guide*.

2. By default, the ibmobjectizer service interval is set to 30 minutes. Set the ibmobjectizer service interval to another value according to your requirement.

For detailed steps, see *Setting up the objectizer service interval* in *IBM Spectrum Scale: Administration Guide*.

3. By default, the identify management mode for unified file and object access is set to `local_mode`. Change the identity management mode according to your requirement.

For detailed steps, see *Configuring authentication and setting identity management modes for unified file and object access* in *IBM Spectrum Scale: Administration Guide*.

For the other unified file and object access tasks that you can do, see *Administering unified file and object access* in *IBM Spectrum Scale: Administration Guide*.

Related concepts

Enabling multi-region object deployment initially

For multi-region object deployment, each region is a separate cluster and on each cluster IBM Spectrum Scale for object storage is installed independently by using the installer.

Installing and using unified file and object access

Unified file and object access are installed when you install IBM Spectrum Scale for object storage in an IBM Spectrum Scale cluster. No additional steps are required for installing unified file and object access.

Chapter 5. Installing IBM Spectrum Scale on AIX nodes

There are three steps for installing GPFS on AIX nodes.

Before you begin installation, read [“Planning for GPFS” on page 219](#) and the [IBM Spectrum Scale FAQ in IBM Documentation](#).

Do not attempt to install GPFS if you do not have the prerequisites listed in [“Hardware requirements” on page 219](#) and [“Software requirements” on page 220](#).

Ensure that the PATH environment variable on each node includes `/usr/lpp/mmfs/bin`.

The installation process includes:

1. [“Creating a file to ease the AIX installation process” on page 453](#)
2. [“Verifying the level of prerequisite software” on page 453](#)
3. [“Procedure for installing GPFS on AIX nodes” on page 453](#)

Creating a file to ease the AIX installation process

Creation of a file that contains all of the nodes in your GPFS cluster prior to the installation of GPFS, will be useful during the installation process. Using either host names or IP addresses when constructing the file will allow you to use this information when creating your cluster through the `mmcrcluster` command.

For example, create the file `/tmp/gpfs.allnodes`, listing the nodes one per line:

```
k145n01.dpd.ibm.com
k145n02.dpd.ibm.com
k145n03.dpd.ibm.com
k145n04.dpd.ibm.com
k145n05.dpd.ibm.com
k145n06.dpd.ibm.com
k145n07.dpd.ibm.com
k145n08.dpd.ibm.com
```

Verifying the level of prerequisite software

Before you can install GPFS, verify that your system has the correct software levels installed.

If your system *does not* have the prerequisite AIX level, refer to the appropriate installation manual before proceeding with your GPFS installation. See the [IBM Spectrum Scale FAQ in IBM Documentation](#) for the latest software levels.

To verify the software version, run the command:

```
WCOLL=/tmp/gpfs.allnodes dsh "oslevel"
```

The system displays the current AIX level.

Procedure for installing GPFS on AIX nodes

These installation procedures are generalized for all levels of GPFS. Ensure you substitute the correct numeric value for the modification (*m*) and fix (*f*) levels, where applicable. The modification and fix level are dependent upon the current level of program support.

Follow these steps to install the GPFS software using the `installp` command:

1. [“Accepting the electronic license agreement” on page 454](#)
2. [“Creating the GPFS directory” on page 454](#)

3. [“Creating the GPFS installation table of contents file” on page 455](#)
4. [“Installing the GPFS man pages” on page 455](#)
5. [“Installing GPFS over a network” on page 455](#)
6. [“Verifying the GPFS installation” on page 455](#)

Accepting the electronic license agreement

The IBM Spectrum Scale software license agreement is shipped and viewable electronically. The electronic license agreement must be accepted before software installation can continue.

IBM Spectrum Scale has following editions based on functional levels:

- IBM Spectrum Scale Standard Edition
- IBM Spectrum Scale Data Access Edition
- IBM Spectrum Scale Advanced Edition
- IBM Spectrum Scale Data Management Edition

For IBM Spectrum Scale Standard Edition or IBM Spectrum Scale Data Access Edition and IBM Spectrum Scale Advanced Edition or IBM Spectrum Scale Data Management Edition installations, the installation cannot occur unless the appropriate license agreements are accepted.

When using the `installp` command, use the `-Y` flag to accept licenses and the `-E` flag to view license agreement files on the media.

GPFS license agreements are retained on the AIX system after installation completes. These license agreements can be viewed after installation in the following directories:

`/usr/swlag/GPFS_standard` (IBM Spectrum Scale Standard Edition or IBM Spectrum Scale Data Access Edition)

`/usr/swlag/GPFS_advanced` (IBM Spectrum Scale Advanced Edition or IBM Spectrum Scale Data Management Edition)

Creating the GPFS directory

To create the GPFS directory:

1. On any node create a temporary subdirectory where GPFS installation images will be extracted. For example:

```
mkdir /tmp/gpfs1pp
```

2. Copy the installation images from the CD-ROM to the new directory, by issuing:

```
bffcreate -qvX -t /tmp/gpfs1pp -d /dev/cd0/AIX all
```

This command places these GPFS installation files in the images directory:

```
gpfs.base
gpfs.docs.data
gpfs.msg.en_US
gpfs.license.xx
gpfs.compression
gpfs.gskit
gpfs.crypto (IBM Spectrum Scale Advanced Edition or IBM Spectrum Scale Data Management Edition only)
gpfs.adv (IBM Spectrum Scale Advanced Edition or IBM Spectrum Scale Data Management Edition only)
```

Creating the GPFS installation table of contents file

To create the GPFS installation table of contents file:

1. Make the new image directory the current directory:

```
cd /tmp/gpfs1pp
```

2. Use the `inutoc .` command to create a `.toc` file. The `.toc` file is used by the `installp` command.

```
inutoc .
```

Installing the GPFS man pages

In order to use the GPFS man pages you must install the `gpfs.docs.data` image.

The GPFS manual pages will be located at `/usr/share/man/`.

Installation consideration: The `gpfs.docs.data` image need not be installed on all nodes if man pages are not desired or local file system space on the node is minimal.

Installing GPFS over a network

Install GPFS according to these directions, where *localNode* is the name of the node on which you are running:

1. If you are installing on a shared file system network, ensure the directory where the GPFS images can be found is NFS exported to all of the nodes planned for your GPFS cluster (`/tmp/gpfs.allnodes`).
2. Ensure an acceptable directory or mountpoint is available on each target node, such as `/tmp/gpfs1pp`. If there is not, create one:

```
WCOLL=/tmp/gpfs.allnodes mmdsh "mkdir /tmp/gpfs1pp"
```

3. If you are installing on a shared file system network, to place the GPFS images on each node in your network, issue:

```
WCOLL=/tmp/gpfs.allnodes mmdsh "mount localNode:/tmp/gpfs1pp /tmp/gpfs1pp"
```

Otherwise, issue:

```
WCOLL=/tmp/gpfs.allnodes mmdsh "rcp localNode:/tmp/gpfs1pp/gpfs* /tmp/gpfs1pp"
WCOLL=/tmp/gpfs.allnodes mmdsh "rcp localNode:/tmp/gpfs1pp/.toc /tmp/gpfs1pp"
```

4. Install GPFS on each node:

```
WCOLL=/tmp/gpfs.allnodes mmdsh "installp -agXYd /tmp/gpfs1pp gpfs"
```

For information about using **rcp** with IBM Spectrum Scale, see [“Remote file copy command” on page 234](#).

Verifying the GPFS installation

Verify that the installation procedure placed the required GPFS files on each node by running the `ls1pp` command on *each* node:

```
ls1pp -l gpfs\*
```

The system returns output similar to the following:

Fileset	Level	State	Description

Path: /usr/lib/objrepos			
gpfs.adv	5.1.5.x	COMMITTED	GPFS Advanced Features
gpfs.base	5.1.5.x	COMMITTED	GPFS File Manager
gpfs.crypto	5.1.5.x	COMMITTED	GPFS Cryptographic Subsystem

gpfs.gskit	8.0.55.x	COMMITTED GPFS GSKit Cryptography Runtime
gpfs.compression	5.1.5.x	COMMITTED GPFS Compression Libraries
gpfs.msg.en_US	5.1.5.x	COMMITTED GPFS Server Messages - U.S. English
gpfs.license.std License	5.1.5.x	COMMITTED IBM Spectrum Scale Standard Edition
Path: /etc/objrepos		
gpfs.base	5.1.5.x	COMMITTED GPFS File Manager
Path: /usr/share/lib/objrepos		
gpfs.docs.data	5.1.5.x	COMMITTED GPFS Server Manpages and Documentation

The output that is returned on your system can vary depending on the edition that you have installed (IBM Spectrum Scale Standard Edition or IBM Spectrum Scale Data Access Edition or IBM Spectrum Scale Advanced Edition or IBM Spectrum Scale Data Management Edition).

Note: The path returned by `ls1pp -l` shows the location of the package control data used by `installp`. The listed path does not show GPFS file locations. To view GPFS file locations, use the `-f` flag.

Chapter 6. Installing IBM Spectrum Scale on Windows nodes

There are several steps for installing GPFS on Windows nodes. The information in this topic points you to the detailed steps.

Before you begin installation, read the following:

- [“Planning for GPFS” on page 219](#)
- The [IBM Spectrum Scale FAQ in IBM Documentation](#)
- [“GPFS for Windows overview” on page 457](#) and all of its subtopics

Do not install GPFS unless you have the prerequisites listed in [“Hardware requirements” on page 219](#) and [“Software requirements” on page 220](#).

The installation process includes:

1. [“Installing GPFS prerequisites” on page 465](#)
2. [“Procedure for installing GPFS on Windows nodes” on page 468](#)
3. [“Configuring a mixed Windows and UNIX \(AIX or Linux\) cluster” on page 470](#)

To install GPFS for Windows, first configure your Windows systems as described in [“Installing GPFS prerequisites” on page 465](#). This includes steps such as joining an Active Directory domain and installing Cygwin from the [Cygwin website \(www.cygwin.com\)](#), which provides a UNIX-like environment on Windows. GPFS installation is simple once the prerequisites are completed. Finally, if your system will be part of a cluster that includes UNIX nodes, follow the steps described in [“Configuring a mixed Windows and UNIX \(AIX or Linux\) cluster” on page 470](#). This includes creating the GPFS Administration service, installing OpenSSH, and other requirements. Complete this process before performing configuration steps common to all GPFS supported platforms.

Note: Throughout this information, UNIX file name conventions are used. For example, the GPFS cluster configuration data is stored in the `/var/mmfs/gen/mmsdrfs` file. On Windows, the UNIX namespace starts under the Cygwin installation directory, which by default is `%SystemDrive%\cygwin64`, so the GPFS cluster configuration data is stored in the `C:\cygwin64\var\mmfs\gen\mmsdrfs` file.

GPFS for Windows overview

GPFS for Windows participates in a new or existing GPFS cluster in conjunction with AIX and Linux systems.

Support includes the following:

- Core GPFS parallel data services
- Windows file system semantics
- A broad complement of advanced GPFS features
- User identity mapping between Windows and UNIX
- Access to GPFS 3.4 and/or later file systems

Identity mapping between Windows and UNIX user accounts is a key feature. System administrators can explicitly match users and groups defined on UNIX with those defined on Windows. Users can maintain file ownership and access rights from either platform. System administrators are not required to define an identity map. GPFS automatically creates a mapping when one is not defined.

GPFS supports the unique semantic requirements posed by Windows. These requirements include case-insensitive names, NTFS-like file attributes, and Windows file locking. GPFS provides a bridge between a Windows and POSIX view of files, while not adversely affecting the functions provided on AIX and Linux.

GPFS for Windows provides the same core services to parallel and serial applications as are available on AIX and Linux. GPFS gives parallel applications simultaneous access to files from any node that has GPFS mounted, while managing a high level of control over all file system operations. System administrators and users have a consistent command interface on AIX, Linux, and Windows. With few exceptions, the commands supported on Windows are identical to those on other GPFS platforms. See [“GPFS limitations on Windows”](#) on page 459 for a list of commands that Windows clients do not support.

Related concepts

Installing GPFS prerequisites

This topic provides details on configuring Windows systems prior to installing GPFS.

Procedure for installing GPFS on Windows nodes

IBM provides GPFS as Windows Installer packages (MSI), which allow both interactive and unattended installations. Perform the GPFS installation steps as the Administrator or some other member of the Administrators group.

Running GPFS commands

Once GPFS is installed, the account you use for GPFS administrative operations (such as creating a cluster and file systems) will depend on your cluster type.

Configuring a mixed Windows and UNIX (AIX or Linux) cluster

Configuring the Windows HPC server

In order for Windows HPC Server to function with GPFS, disable dynamic hosts file and re-enable dynamic DNS updates by doing the following

GPFS support for Windows

The [IBM Spectrum Scale FAQ in IBM Documentation](#) lists the levels of the Windows operating system that are supported by GPFS.

Limited GPFS support for the Windows platform first appeared in GPFS 3.2.1. Subsequent GPFS releases have expanded the set of available functions and features so that now most GPFS capabilities supported on the UNIX platforms are also supported on Windows.

Related concepts

GPFS limitations on Windows

GPFS for Windows supports most of the GPFS features that are available on AIX and Linux, but some limitations apply.

File system name considerations

File name considerations

File names created on UNIX-based GPFS nodes that are not valid on Windows are transformed into valid short names. A name may not be valid either because it is a reserved name like NUL or COM2, or because it has a disallowed character like colon (:) or question mark (?).

Case sensitivity

Native GPFS is case-sensitive; however, Windows applications can choose to use case-sensitive or case-insensitive names.

Antivirus software

If more than one GPFS Windows node is running antivirus software that scans directories and files, shared files only need to be scanned by one GPFS node. It is not necessary to scan shared files more than once.

Differences between GPFS and NTFS

GPFS differs from the Microsoft Windows NT File System (NTFS) in its degree of integration into the Windows administrative environment, Windows Explorer, and the desktop.

Access control on GPFS file systems

GPFS provides support for the Windows access control model for file system objects.

GPFS limitations on Windows

GPFS for Windows supports most of the GPFS features that are available on AIX and Linux, but some limitations apply.

The following limitations apply to configuring and operating a file system on a Windows node:

- File systems must have been created with GPFS 3.1.2.5 or later.
- You cannot upgrade an existing file system that was created with GPFS 3.1 or earlier.

The remaining GPFS for Windows limitations apply only to the Windows nodes in a cluster:

- The following GPFS commands are not supported on Windows:
 - **mmafm*** commands
 - **mmapplypolicy**
 - **mmaudit*** commands
 - **mmbbackup**
 - **mmbbackupconfig, mmrestoreconfig**
 - **mmcall*** commands
 - **mmcdp*** commands
 - **mmces*** commands
 - **mmcheckquota, mmdefedquota, mmedquota, mmlsquota, mmrepquota, mmsetquota**
 - **mmclone**
 - **mmcloud*** commands
 - **mmdelacl, mmeditACL**
 - **mmhadoop*** commands
 - **mmhealth**
 - **mmimgbackup, mmimgrestore**
 - **mmkeyserv**
 - **mmnetverify**
 - **mmnfs*** commands
 - **mmperfmon, mmpmon**
 - **mmrestorefs**
 - **mmsed**
 - **mm smb*** commands
 - **mm snmp*** commands
 - **mm sysmon*** commands
 - **mmuserauth**
 - **spectrumscale**
 - All **GNR** commands
- The GPFS Application Programming Interfaces (APIs) are not supported on Windows.
- The native Windows backup utility is not supported.
- Symbolic links that are created on UNIX-based nodes are specially handled by GPFS Windows nodes; they appear as regular files with a size of 0 and their contents cannot be accessed or modified.
- GPFS on Windows nodes attempts to preserve data integrity between memory-mapped I/O and other forms of I/O on the same computation node. However, if the same file is memory mapped on more than one Windows node, data coherency is not guaranteed between the memory-mapped sections on

these multiple nodes. In other words, GPFS on Windows does not provide distributed shared memory semantics. Therefore, applications that require data coherency between memory-mapped files on more than one node might not function as expected.

- GPFS Windows compute nodes are not supported in AFM clusters.

Related concepts

[GPFS support for Windows](#)

[File system name considerations](#)

[File name considerations](#)

File names created on UNIX-based GPFS nodes that are not valid on Windows are transformed into valid short names. A name may not be valid either because it is a reserved name like NUL or COM2, or because it has a disallowed character like colon (:) or question mark (?).

[Case sensitivity](#)

Native GPFS is case-sensitive; however, Windows applications can choose to use case-sensitive or case-insensitive names.

[Antivirus software](#)

If more than one GPFS Windows node is running antivirus software that scans directories and files, shared files only need to be scanned by one GPFS node. It is not necessary to scan shared files more than once.

[Differences between GPFS and NTFS](#)

GPFS differs from the Microsoft Windows NT File System (NTFS) in its degree of integration into the Windows administrative environment, Windows Explorer, and the desktop.

[Access control on GPFS file systems](#)

GPFS provides support for the Windows access control model for file system objects.

File system name considerations

GPFS file system names should be no longer than 32 characters if the file system will be mounted on a Windows node. GPFS file system names are used as Windows file system labels, and Windows limits the length of these labels to 32 characters. Any attempt to mount a file system with a long name will fail.

You can rename a file system using a command like the following:

```
mmchfs gpfs_file_system_name_that_is_too_long -W gpfs_name_1
```

Related concepts

[GPFS support for Windows](#)

[GPFS limitations on Windows](#)

GPFS for Windows supports most of the GPFS features that are available on AIX and Linux, but some limitations apply.

[File name considerations](#)

File names created on UNIX-based GPFS nodes that are not valid on Windows are transformed into valid short names. A name may not be valid either because it is a reserved name like NUL or COM2, or because it has a disallowed character like colon (:) or question mark (?).

[Case sensitivity](#)

Native GPFS is case-sensitive; however, Windows applications can choose to use case-sensitive or case-insensitive names.

[Antivirus software](#)

If more than one GPFS Windows node is running antivirus software that scans directories and files, shared files only need to be scanned by one GPFS node. It is not necessary to scan shared files more than once.

[Differences between GPFS and NTFS](#)

GPFS differs from the Microsoft Windows NT File System (NTFS) in its degree of integration into the Windows administrative environment, Windows Explorer, and the desktop.

[Access control on GPFS file systems](#)

GPFS provides support for the Windows access control model for file system objects.

File name considerations

File names created on UNIX-based GPFS nodes that are not valid on Windows are transformed into valid short names. A name may not be valid either because it is a reserved name like NUL or COM2, or because it has a disallowed character like colon (:) or question mark (?).

For more information, see [Naming Files, Paths, and Namespaces](#) in Microsoft Windows documentation.

Windows applications can use short names to access GPFS files with Windows file names that are not valid. GPFS generates unique short names using an internal algorithm. You can view these short names using `dir /x` in a DOS command prompt.

Table 43 on page 461 shows an example:

Table 43. Generating short names for Windows	
UNIX	Windows
foo+bar.foobar	FO~23Q_Z.foo
foo bar.-bar	FO~TD}C5._ba
f bar.-bary	F_AMJ5!._ba

Related concepts

[GPFS support for Windows](#)

[GPFS limitations on Windows](#)

GPFS for Windows supports most of the GPFS features that are available on AIX and Linux, but some limitations apply.

[File system name considerations](#)

[Case sensitivity](#)

Native GPFS is case-sensitive; however, Windows applications can choose to use case-sensitive or case-insensitive names.

[Antivirus software](#)

If more than one GPFS Windows node is running antivirus software that scans directories and files, shared files only need to be scanned by one GPFS node. It is not necessary to scan shared files more than once.

[Differences between GPFS and NTFS](#)

GPFS differs from the Microsoft Windows NT File System (NTFS) in its degree of integration into the Windows administrative environment, Windows Explorer, and the desktop.

[Access control on GPFS file systems](#)

GPFS provides support for the Windows access control model for file system objects.

Case sensitivity

Native GPFS is case-sensitive; however, Windows applications can choose to use case-sensitive or case-insensitive names.

This means that case-sensitive applications, such as those using Windows support for POSIX interfaces, behave as expected. Native Win32 applications (such as Windows Explorer) have only case-aware semantics.

The case specified when a file is created is preserved, but in general, file names are case insensitive. For example, Windows Explorer allows you to create a file named **Hello.c**, but an attempt to create **hello.c** in the same folder will fail because the file already exists. If a Windows node accesses a folder that contains two files that are created on a UNIX node with names that differ only in case, Windows inability to distinguish between the two files might lead to unpredictable results.

Related concepts

[GPFS support for Windows](#)

[GPFS limitations on Windows](#)

GPFS for Windows supports most of the GPFS features that are available on AIX and Linux, but some limitations apply.

[File system name considerations](#)

[File name considerations](#)

File names created on UNIX-based GPFS nodes that are not valid on Windows are transformed into valid short names. A name may not be valid either because it is a reserved name like NUL or COM2, or because it has a disallowed character like colon (:) or question mark (?).

[Antivirus software](#)

If more than one GPFS Windows node is running antivirus software that scans directories and files, shared files only need to be scanned by one GPFS node. It is not necessary to scan shared files more than once.

[Differences between GPFS and NTFS](#)

GPFS differs from the Microsoft Windows NT File System (NTFS) in its degree of integration into the Windows administrative environment, Windows Explorer, and the desktop.

[Access control on GPFS file systems](#)

GPFS provides support for the Windows access control model for file system objects.

Antivirus software

If more than one GPFS Windows node is running antivirus software that scans directories and files, shared files only need to be scanned by one GPFS node. It is not necessary to scan shared files more than once.

When you run antivirus scans from more than one node, schedule the scans to run at different times to allow better performance of each scan, as well as to avoid any conflicts that might arise because of concurrent exclusive access attempts by the antivirus software from multiple nodes. Note that enabling real-time antivirus protection for GPFS volumes could significantly degrade GPFS performance and cause excessive resource consumption.

Tip: Consider using a single, designated Windows node to perform all virus scans.

Latest versions of Windows such as Windows 10 come with a built-in antivirus component known as Windows Defender®. While performing real-time scanning of files, Windows Defender might memory-map these files even when they are not in use by any user application. This memory-mapping of files on GPFS file systems by Windows Defender in the background, can result in performance degradation. Therefore, it is recommended that GPFS drives or volumes be excluded from Windows Defender scans all together. Another side effect of this background memory-mapping of GPFS files by antivirus programs is that an attempt to compress or uncompress GPFS files from these Windows nodes might fail. For more information, see *File compression and memory mapping* in *IBM Spectrum Scale: Administration Guide*.

Related concepts

[GPFS support for Windows](#)

[GPFS limitations on Windows](#)

GPFS for Windows supports most of the GPFS features that are available on AIX and Linux, but some limitations apply.

[File system name considerations](#)

[File name considerations](#)

File names created on UNIX-based GPFS nodes that are not valid on Windows are transformed into valid short names. A name may not be valid either because it is a reserved name like NUL or COM2, or because it has a disallowed character like colon (:) or question mark (?).

[Case sensitivity](#)

Native GPFS is case-sensitive; however, Windows applications can choose to use case-sensitive or case-insensitive names.

[Differences between GPFS and NTFS](#)

GPFS differs from the Microsoft Windows NT File System (NTFS) in its degree of integration into the Windows administrative environment, Windows Explorer, and the desktop.

[Access control on GPFS file systems](#)

GPFS provides support for the Windows access control model for file system objects.

Differences between GPFS and NTFS

GPFS differs from the Microsoft Windows NT File System (NTFS) in its degree of integration into the Windows administrative environment, Windows Explorer, and the desktop.

The differences are as follows:

- Manual refreshes are required to see any updates to the GPFS namespace.
- You cannot use the recycle bin.
- You cannot use distributed link tracking. This is a technique through which shell shortcuts and OLE links continue to work after the target file is renamed or moved. Distributed link tracking can help you locate the link sources in case the link source is renamed or moved to another folder on the same or different volume on the same computer, or moved to a folder on any computer in the same domain.
- You cannot use NTFS change journaling. This also means that you cannot use the Microsoft Indexing Service or Windows Search Service to index and search files and folders on GPFS file systems.

GPFS does not support the following NTFS features:

- File compression (on individual files or on all files within a folder)
- Encrypted directories
- Encrypted files (GPFS file encryption is administered through GPFS-specific commands. For more information, see *Encryption* in *IBM Spectrum Scale: Administration Guide*.)
- Quota management (GPFS quotas are administered through GPFS-specific commands)
- Reparse points
- Defragmentation and error-checking tools
- Alternate data streams
- Directory Change Notification
- The assignment of an access control list (ACL) for the entire drive
- Any Access Control Entry (ACE) type other than ALLOW, DENY, AUDIT and ALARM.

Upon an attempt to set any unsupported ACE type (such as OBJECT or CALLBACK/Conditional type), GPFS will filter and skip the unsupported ACE type from the ACL, resulting in a lossy ACL getting applied to the GPFS file or directory. Since conditional ACEs get filtered, GPFS does not support Dynamic Access Control (DAC)

- Generation of AUDIT and ALARM events specified in a System Access Control List (SACL). GPFS is capable of storing SACL content, but will not interpret it.
- The scanning of all files or directories that a particular SID owns (**FSCTL_FIND_FILES_BY_SID**)
- Windows sparse files API
- A change journal for file activity
- Transactional NTFS (also known as TxF)

Related concepts

[GPFS support for Windows](#)

[GPFS limitations on Windows](#)

GPFS for Windows supports most of the GPFS features that are available on AIX and Linux, but some limitations apply.

[File system name considerations](#)

[File name considerations](#)

File names created on UNIX-based GPFS nodes that are not valid on Windows are transformed into valid short names. A name may not be valid either because it is a reserved name like NUL or COM2, or because it has a disallowed character like colon (:) or question mark (?).

Case sensitivity

Native GPFS is case-sensitive; however, Windows applications can choose to use case-sensitive or case-insensitive names.

Antivirus software

If more than one GPFS Windows node is running antivirus software that scans directories and files, shared files only need to be scanned by one GPFS node. It is not necessary to scan shared files more than once.

Access control on GPFS file systems

GPFS provides support for the Windows access control model for file system objects.

Access control on GPFS file systems

GPFS provides support for the Windows access control model for file system objects.

Each GPFS file or directory has a Security Descriptor (SD) object associated with it and you can use the standard Windows interfaces for viewing and changing access permissions and object ownership (for example, Windows Explorer Security dialog panel). Internally, a Windows SD is converted to an NFS V4 access control list (ACL) object, which ensures that access control is performed consistently on other supported operating systems. GPFS supports all discretionary access control list (DACL) operations, including inheritance. GPFS is capable of storing system access control list (SACL) objects, but generation of AUDIT and ALARM events specified in SACL contents is not supported.

An important distinction between GPFS and Microsoft Windows NT File Systems (NTFS) is the default set of permissions for the root (top-level) directory on the file system. On a typical NTFS volume, the DACL for the top-level folder has several inheritable entries that grant full access to certain special accounts, as well as some level of access to nonprivileged users. For example, on a typical NTFS volume, the members of the local group **Users** would be able to create folders and files in the top-level folder. This approach differs substantially from the traditional UNIX convention where the root directory on any file system is only writable by the local **root** superuser by default. GPFS adheres to the latter convention; the root directory on a new file system is only writable by the UNIX user **root**, and does not have an extended ACL when the file system is created. This is to avoid impacting performance in UNIX-only environments, where the use of extended ACLs is not common.

When a new GPFS file system is accessed from a Windows client for the first time, an immutable security descriptor object is created for the root directory automatically. This immutable security descriptor for the root directory contains a non-inheritable DACL that grants full access to the local **Administrators** group and read-only access to the **Everyone** group. This allows only privileged Windows users (members of the local **Administrators** group) to create new files and folders immediately under the root directory. Because the root directory DACL has no inheritable entries, new top-level objects under the root directory are created with a default non-inheritable DACL that only grants local **Administrators** and **SYSTEM** accounts full access. Therefore, as a best practice, privileged Windows users must create top-level objects under the root directory and then explicitly set inheritable DACLs on these top-level directories as appropriate (granting the necessary level of access to non-privileged users).

Note: Some applications expect to find NTFS-style permissions on all file systems and they might not function properly when that is not the case. Running such an application in a GPFS folder where permissions have been set similar to NTFS defaults might correct this.

Related concepts

GPFS support for Windows

GPFS limitations on Windows

GPFS for Windows supports most of the GPFS features that are available on AIX and Linux, but some limitations apply.

File system name considerations

File name considerations

File names created on UNIX-based GPFS nodes that are not valid on Windows are transformed into valid short names. A name may not be valid either because it is a reserved name like NUL or COM2, or because it has a disallowed character like colon (:) or question mark (?).

Case sensitivity

Native GPFS is case-sensitive; however, Windows applications can choose to use case-sensitive or case-insensitive names.

Antivirus software

If more than one GPFS Windows node is running antivirus software that scans directories and files, shared files only need to be scanned by one GPFS node. It is not necessary to scan shared files more than once.

Differences between GPFS and NTFS

GPFS differs from the Microsoft Windows NT File System (NTFS) in its degree of integration into the Windows administrative environment, Windows Explorer, and the desktop.

Installing GPFS prerequisites

This topic provides details on configuring Windows systems prior to installing GPFS.

Perform the following steps:

1. [“Configuring Windows” on page 466](#)
 - a. [“Assigning a static IP address” on page 466](#)
 - b. [“Joining an Active Directory domain” on page 466](#)
 - c. [“Disabling the Windows firewall” on page 466](#)
 - d. [“Installing the Tracefmt and Tracelog programs \(optional\)” on page 466](#)
2. From here on, all GPFS installation steps must be executed as a special user with Administrator privileges. Further, all GPFS administrative operations (such as creating a cluster and file systems) must also be issued as the same user. This special user will depend on your cluster type. For Windows homogeneous clusters, run GPFS commands as a member of the Domain Admins group or as a domain account that is a member of the local Administrators group. For clusters with both Windows and UNIX nodes, run GPFS commands as root, a special domain user account described in [“Creating the GPFS administrative account” on page 471](#).
3. [“Installing Cygwin” on page 466](#)

See the [IBM Spectrum Scale FAQ in IBM Documentation](#) for the latest:

- Software recommendations
- Configuration information

For additional requirements when setting up clusters that contain both Windows nodes and AIX or Linux nodes, see [“Configuring a mixed Windows and UNIX \(AIX or Linux\) cluster” on page 470](#).

Related concepts

GPFS for Windows overview

GPFS for Windows participates in a new or existing GPFS cluster in conjunction with AIX and Linux systems.

Procedure for installing GPFS on Windows nodes

IBM provides GPFS as Windows Installer packages (MSI), which allow both interactive and unattended installations. Perform the GPFS installation steps as the Administrator or some other member of the Administrators group.

Running GPFS commands

Once GPFS is installed, the account you use for GPFS administrative operations (such as creating a cluster and file systems) will depend on your cluster type.

Configuring a mixed Windows and UNIX (AIX or Linux) cluster

Configuring the Windows HPC server

In order for Windows HPC Server to function with GPFS, disable dynamic hosts file and re-enable dynamic DNS updates by doing the following

Configuring Windows

This topic provides some details on installing and configuring Windows on systems that will be added to a GPFS cluster.

The [IBM Spectrum Scale FAQ in IBM Documentation](#) lists the versions of the Windows operating system that are supported by GPFS.

Assigning a static IP address

GPFS communication requires invariant static IP addresses for each GPFS node.

Joining an Active Directory domain

All Windows systems in the same GPFS cluster should be members of the same Active Directory domain. Join these systems to the Windows domain before adding them to a GPFS cluster.

GPFS expects that all Windows nodes in a cluster are members of the same domain. This gives domain users a consistent identity and consistent file access rights independent of the system they are using. The domain controllers, which run the Active Directory Domain Services, are not required to be members of the GPFS cluster.

Refer to your Windows Server documentation for information on how to install and administer Active Directory Domain Services.

Disabling the Windows firewall

GPFS requires that you modify the default Windows Firewall settings.

The simplest change that will allow GPFS to operate properly is to disable the firewall. Open **Windows Firewall** in the Control Panel and click **Turn Windows firewall on or off**, and select **Off** under the General tab. For related information, see *GPFS port usage* in *IBM Spectrum Scale: Administration Guide*.

Installing the Tracefmt and Tracelog programs (optional)

GPFS diagnostic tracing (mmtracectl) on Windows uses the Microsoft programs called tracefmt.exe and tracelog.exe. These programs are not included with Windows but can be downloaded from Microsoft. The tracefmt.exe and tracelog.exe programs are only for tracing support and are not required for normal GPFS operations.

The tracefmt.exe and tracelog.exe programs are included with the Windows Driver Kit (WDK) as well as the Windows SDK. You can download either of these packages from the [Microsoft Download Center \(www.microsoft.com/download\)](#).

To allow GPFS diagnostic tracing on Windows using the WDK, follow these steps:

1. Download the latest version of the Windows Driver Kit (WDK) from Microsoft.
2. Install the Tools feature of the WDK on some system to obtain a copy of tracefmt.exe and tracelog.exe.
3. Locate and copy tracefmt.exe and tracelog.exe to both the /usr/lpp/mmfs/bin and /usr/lpp/mmfs/win directories to ensure mmtracectl properly locates these programs.

For additional information about GPFS diagnostic tracing, see *The GPFS trace facility* in *IBM Spectrum Scale: Problem Determination Guide*.

Installing Cygwin

Cygwin is a POSIX environment available for Windows and can be downloaded from the [Cygwin website \(www.cygwin.com\)](#). GPFS uses this component to support many of its administrative scripts. System

administrators have the option of using either the GPFS Admin Command Prompt or GPFS Admin Korn Shell to run GPFS commands.

Cygwin must be installed before installing GPFS. It is a software package that provides a Unix-like environment on Windows and provides runtime support for POSIX applications and includes programs such as `grep`, `ksh`, `ls`, and `ps`.

When running Cygwin setup, only the standard packages are installed by default. GPFS requires installation of additional packages, which are listed in [“Installing the 64-bit version of Cygwin”](#) on page 467.

Note: Starting with GPFS 4.1.1, the 32-bit version of Cygwin is no longer supported for Windows nodes running GPFS. Users that are running GPFS 4.1 with the 32-bit version of Cygwin installed must upgrade to the 64-bit version of Cygwin before installing GPFS 4.1.1. For more information, see [“Upgrading from the 32-bit version of Cygwin to the 64-bit version of Cygwin”](#) on page 467. For users on GPFS releases prior to 4.1 (SUA based), see [“Offline upgrade with complete cluster shutdown”](#) on page 590.

Upgrading from the 32-bit version of Cygwin to the 64-bit version of Cygwin

Follow these instructions to upgrade from the 32-bit version of Cygwin to the 64-bit version of Cygwin:

1. Uninstall GPFS 4.1 and reboot.
2. Uninstall IBM GSKit for GPFS and reboot.
3. Uninstall the GPFS 4.1 license.
4. Stop and delete any Cygwin 32-bit services, such as OpenSSH, that might have been configured.
5. Do *not* uninstall the 32-bit version of Cygwin yet, or you may lose GPFS configuration information.
6. Install the 64-bit version of Cygwin using the instructions in [“Installing the 64-bit version of Cygwin”](#) on page 467.
7. Install the GPFS 4.1.1 license for the appropriate edition; for example, `gpfs.ext-4.1.1-Windows-license.msi`.
8. Install the appropriate GPFS 4.1.1 edition; for example, `gpfs.ext-4.1.1.x-Windows.msi`.
9. Install IBM GSKit for GPFS.
10. Uninstall the 32-bit version of Cygwin completely.
11. Follow the procedures in [“Installing and configuring OpenSSH on Windows nodes”](#) on page 472.

Installing the 64-bit version of Cygwin

To install the 64-bit version of Cygwin for Windows, follow these steps:

1. Logon to the Windows node as the account you will use for GPFS administrative operations.
2. Go to the [Cygwin website \(www.cygwin.com\)](http://www.cygwin.com), and click the **Install Cygwin** link in the upper-left pane.
3. Download and start the installation of the `setup-x86_64.exe` file.
4. Follow the prompts to continue with the install. The default settings are recommended until the Select Packages dialog is displayed. Then, select the following packages (use the Search field to quickly find these packages):
 - `diffutils`: A GNU collection of diff utilities
 - `flip`: Convert text file line endings between Unix and DOS formats
 - `m4`: GNU implementation of the traditional Unix macro processor
 - `mksh`: MirBSD Korn Shell
 - `perl`: Perl programming language interpreter
 - `procps-ng`: System and process monitoring utilities
 - `openssh`: The OpenSSH server and client programs (only required if you plan on mixing Linux, AIX and Windows nodes in the same cluster)

5. Click the **Next** button, and continue to follow the prompts to complete the installation.
6. If you are using a mixed cluster environment (Windows and Linux/AIX), follow the steps in [“Installing and configuring OpenSSH on Windows nodes”](#) on page 472.

Procedure for installing GPFS on Windows nodes

IBM provides GPFS as Windows Installer packages (MSI), which allow both interactive and unattended installations. Perform the GPFS installation steps as the Administrator or some other member of the Administrators group.

Before installing GPFS on Windows nodes, verify that all the installation prerequisites have been met. For more information, see [“Installing GPFS prerequisites”](#) on page 465.

To install GPFS on a Windows node, follow these steps:

1. Run the following license installation package from the product media and accept the license:

`gpfs.base-5.1.x.x-Windows-license.msi` (IBM Spectrum Scale Standard Edition or IBM Spectrum Scale Data Access Edition)

Rebooting is normally not required.

2. Download and install the latest service level of GPFS from the [IBM Support Portal: Downloads for General Parallel File System](#) (www.ibm.com/support/entry/portal/Downloads/Software/Cluster_software/General_Parallel_File_System). The GPFS package name includes the GPFS version (for example, `gpfs.base-5.1.x.x-Windows.msi`). For the latest information on the supported Windows operating system versions, see the [IBM Spectrum Scale FAQ in IBM Documentation](#).

Under some circumstances, the installation process will prompt you to reboot the systems when it is required. You do not need to install the GPFS 5.1.x.x package included on the media before installing the latest update.

3. Download and install the latest level of IBM GSKit for GPFS. GSKit (`gpfs.gskit-x.x.x.x.msi`) comes as a single package that works with the IBM Spectrum Scale Standard Edition or IBM Spectrum Scale Data Access Edition.

Note: For this release, the IBM GSKit version must be at least 8.0.55.x or higher.

To upgrade GPFS, do the following:

1. Uninstall your current GPFS packages (for example, `gpfs.base-5.x.x.0-Windows.msi`, `gpfs.ext-5.x.x.0-Windows.msi`, and `gpfs.gskit-8.0.x.x.msi`).
2. Reboot.
3. Install the latest PTF packages (for example, `gpfs.base-5.1.x.x-Windows.msi`).

When upgrading, you do not need to uninstall and reinstall the license package unless you are explicitly instructed to do so by IBM. In addition, `gpfs.gskit` only needs to be upgraded if the update (zip file) contains a newer version.

For more information, refer to [Chapter 3, “Steps for establishing and starting your IBM Spectrum Scale cluster,”](#) on page 349.

The GPFS installation package provides some options that you can select by installing GPFS from a Command Prompt. Property values control the options. The command line syntax is:

```
msiexec.exe /package <Package> [Optional Parameters] [Property=Value]
```

The following property values are available:

AgreeToLicense=yes

This property applies to the license package `gpfs.base-5.1.x.x-Windows-license.msi`. It allows GPFS to support an installation procedure that does not require user input. IBM recommends that you perform at least one interactive installation on a typical system before attempting an

unattended installation. This will help identify any system configuration issues that could cause the installation to fail.

The following command installs GPFS without prompting you for input or generating any error dialog boxes:

```
msiexec /package gpfs.base-5.1.x.x-Windows-license.msi /passive AgreeToLicense=yes
```

The `msiexec.exe` executable file supports display options other than `/passive`. See `msiexec` documentation for details.

RemoteShell=no

This property applies to the product package (for example, `gpfs.base-5.1.x.x-Windows-license.msi`). It is equivalent to running `mmwinervctl set -remote-shell no`, but performs this configuration change before `mmwinerv` initially starts. This option is available to satisfy security policies that restrict network communication protocols.

You can verify that GPFS is installed correctly by opening the Programs and Features control panel. **IBM Spectrum Scale** should be included in the list of installed programs. The program's version should match the version of the update package.

The GPFS software license agreement is shipped and is viewable electronically. The license agreement will remain available in the `%SystemDrive%\Program Files\IBM\GPFS\5.1.5.x\license` directory for future access.

Related concepts

[GPFS for Windows overview](#)

GPFS for Windows participates in a new or existing GPFS cluster in conjunction with AIX and Linux systems.

[Installing GPFS prerequisites](#)

This topic provides details on configuring Windows systems prior to installing GPFS.

[Running GPFS commands](#)

Once GPFS is installed, the account you use for GPFS administrative operations (such as creating a cluster and file systems) will depend on your cluster type.

[Configuring a mixed Windows and UNIX \(AIX or Linux\) cluster](#)

[Configuring the Windows HPC server](#)

In order for Windows HPC Server to function with GPFS, disable dynamic hosts file and re-enable dynamic DNS updates by doing the following

Running GPFS commands

Once GPFS is installed, the account you use for GPFS administrative operations (such as creating a cluster and file systems) will depend on your cluster type.

For Windows clusters, run GPFS commands as a member of the `Domain Admins` group or as a domain account that is part of the local `Administrators` group. For clusters with both Windows and UNIX nodes, run GPFS commands as `root`, a special domain user account described in [“Creating the GPFS administrative account”](#) on page 471.

You must run all GPFS commands from either a GPFS Admin Command Prompt or a GPFS Admin Korn Shell. The GPFS administrative commands may not work properly if issued from the Cygwin Terminal. Open a new GPFS Admin Command Prompt or GPFS Admin Korn Shell after installing GPFS so that it uses the updated `PATH` environment variable required to execute GPFS commands.

GPFS Admin Command Prompt and GPFS Admin Korn Shell can be accessed using their desktop shortcuts or from under the IBM GPFS Start menu group. Each of these program links starts its respective shell using the `Run as Administrator` option.

Related concepts

[GPFS for Windows overview](#)

GPFS for Windows participates in a new or existing GPFS cluster in conjunction with AIX and Linux systems.

Installing GPFS prerequisites

This topic provides details on configuring Windows systems prior to installing GPFS.

Procedure for installing GPFS on Windows nodes

IBM provides GPFS as Windows Installer packages (MSI), which allow both interactive and unattended installations. Perform the GPFS installation steps as the Administrator or some other member of the Administrators group.

Configuring a mixed Windows and UNIX (AIX or Linux) cluster

Configuring the Windows HPC server

In order for Windows HPC Server to function with GPFS, disable dynamic hosts file and re-enable dynamic DNS updates by doing the following

Configuring a mixed Windows and UNIX (AIX or Linux) cluster

For GPFS clusters that include both Windows and UNIX (AIX or Linux) nodes, this topic describes the additional configuration steps needed beyond those described in [“Installing GPFS prerequisites” on page 465](#).

For mixed clusters, perform the following steps:

1. Optionally, install and configure identity mapping on your Active Directory domain controller (see [“Identity Mapping for Unix \(IDMU\) / RFC 2307 Attributes” on page 470](#)).
2. Create the root administrative account (see [“Creating the GPFS administrative account” on page 471](#)).
3. Edit the Domain Group Policy to give root the right to log on as a service (see [“Allowing the GPFS administrative account to run as a service” on page 471](#)).
4. Configure the GPFS Administration service (mmwinsserv) to run as root (see [“Configuring the GPFS Administration service” on page 472](#)).
5. Install and configure OpenSSH (see [“Installing and configuring OpenSSH on Windows nodes” on page 472](#)).

Complete this process before performing configuration steps common to all GPFS supported platforms.

Identity Mapping for Unix (IDMU) / RFC 2307 Attributes

GPFS can exploit a Windows Server feature called RFC 2307 attributes to provide consistent identities among all nodes in a cluster.

GPFS expects that all Windows nodes in a cluster are members of the same Active Directory domain. This gives domain users a consistent identity and consistent file access rights independent of the system they are using.

GPFS can exploit the RFC 2307 attributes for mapping users and groups between Windows and UNIX. These attributes can be administered by using Identity Mapping for Unix (IDMU) from Microsoft in Windows Server versions up to and including Windows Server 2012 R2. Beginning with Windows Server 2016 (since the IDMU MMC-snap-in has been removed) you can specify these RFC 2307 attributes by using the **Active Directory Users and Computers MMC Snap-in** or the **Active Directory Administrative Center**. 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 Spectrum 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.

For more information, see *Identity management on Windows / RFC 2307 Attributes* instructions on editing IDMU or RFC 2307 attributes in *IBM Spectrum Scale: Administration Guide*.

GPFS uses RFC 2307 attributes, specifically `uidNumber`, to map users and `gidNumber` to map groups between Windows and Unix. From the perspective of GPFS, IDMU is synonymous with RFC 2307 attributes. This means that any references to IDMU in the documentation should be interpreted as RFC 2307 attributes.

For IDMU or ADUC installation and configuration information, see *Configuring ID mappings in Active Directory Users and Computers for Active Directory Users and Computers (ADUC) in IBM Spectrum Scale: Administration Guide*.

Creating the GPFS administrative account

GPFS uses an administrative account in the Active Directory domain named **root** in order to interoperate with UNIX nodes in the cluster. Create this administrative account as follows:

1. Create a domain user with the logon name **root**.
2. Add user **root** to the **Domain Admins** group or to the local Administrators group on each Windows node.
3. In **root** Properties/Profile/Home/LocalPath, define a HOME directory such as `C:\Users\root\home` that does not include spaces in the path name and is not the same as the profile path.
4. Give **root** the right to log on as a service as described in [“Allowing the GPFS administrative account to run as a service”](#) on page 471.

Step “3” on page 471 is required for the Cygwin environment (described in [“Installing Cygwin”](#) on page 466) to operate correctly. Avoid using a path that contains a space character in any of the path names. Also avoid using **root**'s profile path (for example, `C:\User\root`). OpenSSH requires specific permissions on this directory, which can interfere with some Windows applications.

You may need to create the HOME directory on each node in your GPFS cluster. Make sure that **root** owns this directory.

Allowing the GPFS administrative account to run as a service

Clusters that depend on a **root** account to interoperate with UNIX nodes in a cluster will need to configure the GPFS Administrative Service (`mmwinsserv`) to run as the **root** account. For this, **root** needs to be assigned the right to log on as a service. See [“Configuring the GPFS Administration service”](#) on page 472 for details.

The right to log on as a service is controlled by the Local Security Policy of each Windows node. You can use the Domain Group Policy to set the Local Security Policy on all Windows nodes in a GPFS cluster.

The following procedure assigns the *log on as a service* right to an account when the domain controller is running on Windows Server 2008:

1. Open **Group Policy Management** (available under **Administrative Tools**).
2. In the console tree, expand **Forest name/Domains/Domain name/Group Policy Objects**.
3. Right click **Default Domain Policy** and select **Edit**.
4. In the console tree of the Group Policy Management Editor, expand down to **Computer Configuration/Policies/Windows Settings/Security Settings/Local Policies/User Rights Assignment**.
5. Double click the **Log on as a service policy**.
6. Check **Define these policy settings if necessary**.
7. Use **Add User or Group...** to include the **DomainName\root** account in the policy, then click **OK**.

Refer to your *Windows Server* documentation for a full explanation of Local Security Policy and Group Policy Management.

Configuring the GPFS Administration service

GPFS for Windows includes a service called `mmwinerv`. In the Windows Services management console, this service has the name GPFS Administration. `mmwinerv` supports GPFS operations such as **autoload** and remote command execution in Windows GPFS clusters. The Linux and AIX versions of GPFS do not have a similar component. The `mmwinerv` service is used on all Windows nodes starting with GPFS 3.3.

The GPFS installation package configures `mmwinerv` to run using the default **LocalSystem** account. This account supports Windows GPFS clusters. For clusters that include both Windows and UNIX nodes, you must configure `mmwinerv` to run as **root**, the GPFS administrative account. Unlike **LocalSystem**, **root** can access the IDMU service and can access other GPFS nodes as required by some cluster configurations.

For IDMU installation and configuration information, see *Identity management on Windows in IBM Spectrum Scale: Administration Guide*. For information on supporting administrative access to GPFS nodes, see the *Requirements for administering a GPFS file system* topic in the *IBM Spectrum Scale: Administration Guide*.

Before configuring `mmwinerv` to run as **root**, you must first grant **root** the right to run as a service. For details, see [“Allowing the GPFS administrative account to run as a service”](#) on page 471.

Use the GPFS command `mmwinervctl` to set and maintain the GPFS Administration service's log on account. `mmwinervctl` must be run on a Windows node. You can run `mmwinervctl` to set the service account before adding Windows nodes to a cluster. You can also use this command to change or update the account on nodes that are already in a cluster. GPFS can be running or stopped when executing `mmwinervctl`, however, refrain from running other GPFS administrative commands at the same time.

In this example, `mmwinervctl` configures three nodes before they are added to a GPFS cluster containing both Windows and UNIX:

```
mmwinervctl set -N node1,node2,node3 --account mydomain/root --password mypwd --remote-shell no
```

Whenever `root`'s password changes, the `mmwinerv` logon information needs to be updated to use the new password. The following command updates on all Windows nodes in a cluster with a new password:

```
mmwinervctl set -N all --password mynewpwd
```

As long as `mmwinerv` is running, the service will not be affected by an expired or changed password and GPFS will continue to function normally. However, GPFS will not start after a system reboot when `mmwinerv` is configured with an invalid password. If for any reason the Windows domain or `root` password changes, then `mmwinervctl` should be used to update the domain and password. The domain and password can also be updated on a per node basis by choosing Administrative Tools > Computer Management > Services and Applications > Services, and selecting GPFS Administration. Choose File > Properties > Logon and update the `<domain>\username` and the password.

For more information, see **`mmwinervctl` command** in *IBM Spectrum Scale: Command and Programming Reference*.

Installing and configuring OpenSSH on Windows nodes

If using a mixed cluster, OpenSSH must be configured on the Windows nodes. Refer to the [Cygwin FAQ \(www.cygwin.com/faq.html\)](http://www.cygwin.com/faq.html) and documentation on how to setup `sshd`. Replace the usage of the account `cyg_server` in the Cygwin documentation with `root` when setting up a privileged account for `sshd`.

The following are some guidelines in addition to the Cygwin instructions on setting up `sshd`:

1. Verify that all nodes can be pinged among themselves by host name, Fully Qualified Domain Name (FQDN) and IP address.
2. If not using IPv6, disable it. For more information, see [How to disable IPv6 or its components in Windows \(support.microsoft.com/kb/929852\)](http://support.microsoft.com/kb/929852).

3. Check that passwd contains the privileged user that you plan to use for GPFS operations, as well as its correct home path:

```
$ cat /etc/passwd | grep "root"
```

```
root:unused:11103:10513:U-WINGPFS\root,S-1-5-21-3330551852-1995197583-3793546845-1103:/cygdrive/c/home/  
root:/bin/bash
```

If the user is not listed, rebuild your passwd:

```
mkpasswd -l -d wingpfs > /etc/passwd
```

4. From the Cygwin shell, run `/usr/bin/ssh-host-config` and respond yes to the prompts. When prompted to enter the value of CYGWIN for the daemon, enter `ntsec`. Specify `root` in response to the query for the new user name. You may receive the following warning:

```
***Warning: The specified account 'root' does not have the  
***Warning: required permissions or group memberships. This may  
***Warning: cause problems if not corrected; continuing...
```

As long as the account (in this case, `root`) is in the local Administrators group, you can ignore this warning.

5. When the installation is complete, enter the following:

```
$ net start sshd
```

```
The CYGWIN sshd service is starting.  
The CYGWIN sshd service was started successfully.
```

Note: The OpenSSH READMEs are available at `/usr/share/doc/openssh`. Also see the [IBM Spectrum Scale FAQ in IBM Documentation](#).

Once OpenSSH is installed, the GPFS administrative account `root` needs to be configured so that it can issue `ssh` and `scp` commands without requiring a password and without producing any extraneous messages. This kind of passwordless access is required from any node used for GPFS administration to all other nodes in the cluster.

For additional information, see *Requirements for administering a GPFS file system* in *IBM Spectrum Scale: Administration Guide* and *Troubleshooting Windows problems* in *IBM Spectrum Scale: Problem Determination Guide*.

Related concepts

[GPFS for Windows overview](#)

GPFS for Windows participates in a new or existing GPFS cluster in conjunction with AIX and Linux systems.

[Installing GPFS prerequisites](#)

This topic provides details on configuring Windows systems prior to installing GPFS.

[Procedure for installing GPFS on Windows nodes](#)

IBM provides GPFS as Windows Installer packages (MSI), which allow both interactive and unattended installations. Perform the GPFS installation steps as the Administrator or some other member of the Administrators group.

[Running GPFS commands](#)

Once GPFS is installed, the account you use for GPFS administrative operations (such as creating a cluster and file systems) will depend on your cluster type.

[Configuring the Windows HPC server](#)

In order for Windows HPC Server to function with GPFS, disable dynamic hosts file and re-enable dynamic DNS updates by doing the following

Configuring the Windows HPC server

In order for Windows HPC Server to function with GPFS, disable dynamic hosts file and re-enable dynamic DNS updates by doing the following

1. On all nodes in the Windows HPC cluster, open the hosts file %systemroot%\system32\drivers\etc\hosts and change `ManageFile = true` to `ManageFile = false`.
2. On the HPC head node, execute the following from HPC PowerShell, assuming all nodes are part of the head node's Active Directory domain.

Go to Start > All Programs > Microsoft HPC Pack > HPC PowerShell and execute:

```
Set-HpcNetwork -EnterpriseDnsRegistrationType WithConnectionDnsSuffix
```

Related concepts

[GPFS for Windows overview](#)

GPFS for Windows participates in a new or existing GPFS cluster in conjunction with AIX and Linux systems.

[Installing GPFS prerequisites](#)

This topic provides details on configuring Windows systems prior to installing GPFS.

[Procedure for installing GPFS on Windows nodes](#)

IBM provides GPFS as Windows Installer packages (MSI), which allow both interactive and unattended installations. Perform the GPFS installation steps as the Administrator or some other member of the Administrators group.

[Running GPFS commands](#)

Once GPFS is installed, the account you use for GPFS administrative operations (such as creating a cluster and file systems) will depend on your cluster type.

[Configuring a mixed Windows and UNIX \(AIX or Linux\) cluster](#)

Chapter 7. Installing Cloud services on IBM Spectrum Scale nodes

The Cloud services installation is not included in the IBM Spectrum Scale installation toolkit workflow. It needs to be installed separately on top of the IBM Spectrum Scale cluster. However, the Cloud services RPMs are available with the IBM Spectrum Scale package.

Before you begin, ensure that you have created a node class. For more information, see the *Creating a node class* topic in the *IBM Spectrum Scale: Administration Guide*. Also, ensure that your server meets the required prerequisites. For more information, see [“Software requirements for Cloud services” on page 329](#)

Creating a user-defined node class for Transparent cloud tiering or Cloud data sharing

This topic provides step-by-step instructions for creating and configuring the required node class to specify which IBM Spectrum Scale nodes are used for Cloud services.

Note: You need to run this command only once on any node on the IBM Spectrum Scale cluster.

You can install Cloud services on a maximum of four nodes per node group (any combination of NSD or CES nodes) on the IBM Spectrum Scale cluster (which can have a maximum of four node groups). Before you install Cloud services, you need to create a node class that specifies the IP addresses (or fully qualified host names) of the nodes where the server packages are going to be installed. You can use any host name that the `-N` option accepts. For details, see *mmcrnodeclass command* in *IBM Spectrum Scale: Command and Programming Reference*.

You can enable and manage independent groups of Cloud services nodes in different node classes for use with different network configurations per node class. Nodes are mutually exclusive to each node class and cannot be shared with another node class that has Cloud services nodes. For example, the node class `TCTNodeClass1` has `node1` and `node2` and the node class `TCTNodeClass2` has `node3` and `node4`. All nodes are Cloud services enabled nodes. In this case, the node class `TCTNodeClass2` cannot enable `node1` as a Cloud services node under the `TCTNodeClass2` node class because it is already marked for Cloud services usage under the `TCTNodeClass1` node class. Additionally, each group of Cloud services enabled nodes under a node class cannot share file system with Cloud services enabled nodes from a different node class. And each Cloud services enabled node under a node class is limited to use only one common file system with each node in that node class. Therefore, you must manage each group of Cloud services nodes within a node class as a pool of nodes that require a single common file system.

Note: It is recommended that you set up at least two nodes so that you have good availability for the service if a node were to go down. Ensure to use the GPFS cluster IP address that gets displayed when you run the `mmfscluster` command.

To create a node class, issue a command according to this syntax:

```
mmcrnodeclass ClassName -N {Node[,Node...]} | NodeFile | NodeClass}
```

For example, to create a node class called `TCTNodeClass` by using three nodes `10.10.10.10`, `11.11.11.11`, and `12.12.12.12`, issue this command:

```
mmcrnodeclass TCTNodeClass -N 10.10.10.10,11.11.11.11,12.12.12.12
```

To verify that the node class is created, issue this command:

```
mmnlsnodeclass
```

The system displays output similar to this output:

Node	Class	Name	Members

TCTNodeClass		node10,node11,node12	

Next step: [“Installation steps” on page 476](#)

Installation steps

This topic describes the procedure for installing Cloud services on IBM Spectrum Scale nodes.

Cloud services creates the following folders after installation:

- /var/MCStore
- <filesystem>/mcstore
- /opt/ibm/MCStore

Do not make any changes to these folders in order to avoid any undesired results.

The Cloud services RPMs are available with IBM Spectrum Scale Advanced Edition, IBM Spectrum Scale Data Management Edition, IBM Spectrum Scale Developer Edition, or IBM Spectrum Scale Erasure Code Edition. You can verify this by issuing this command: **rpm -qa | grep gpfs.adv**. Depending on the target Linux distribution, the RPMs are available in the following paths where *_rpms or *_debs is the component specific directory:

- Red Hat Enterprise Linux 7.x: /usr/lpp/mmfs/5.0.x.x/*_rpms/rhel7
- Ubuntu 16.04 LTS: /usr/lpp/mmfs/5.0.x.x/*_debs/ubuntu16
- SLES 12: /usr/lpp/mmfs/5.0.x.x/*_rpms/sles12

With each build of IBM Spectrum Scale, two RPMs are available for Cloud services:

- gpfs.tct.server-x.x.x.x86_64.rpm. This is the server package that is installed on the Cloud services nodes. These nodes must have outbound WAN connectivity to be able to support data migration between IBM Spectrum Scale cluster and the cloud object storage. When installed on non-cloud gateway nodes, files are recalled locally without routing the requests through the gateway nodes. To enable the client-assisted recall, you must run **mmcloudgateway clientassist enable**.
- gpfs.tct.client-x.x.x.x86_64.rpm. This is the client package that contains the Cloud services binary files (specifically the data manager commands). This package must be installed on all remaining cluster nodes to activate ILM-based file migration or file recall that is initiated from that node (every node in the cluster is safe). The client RPM is not required on the nodes where the server package is installed.

Note: It is important to install the client package on every node so that files that are deleted from the IBM Spectrum Scale cluster are deleted efficiently from the cloud storage tier. If this package is not installed, files are deleted less aggressively and only as a result of running the reconcile service.

Note: For policy-based migrations, it is recommended that the file system manager node (rather all nodes with manager roles) is installed at least with the client RPM. This ensures that if any migration request goes to the file system manager node that is not a Transparent cloud tiering node, the request is forwarded to one of the server nodes in the node class, and the migration seamlessly takes place. If the RPM is not there on the file system manager, then migration fails. Transparent cloud tiering is needed on all clients (not just for migration requests) or transparent recall does not work properly and delete processing is not handled efficiently for files transferred to the cloud.

1. Copy the server RPMs to the desired nodes.
2. Install Cloud services package on each node by issuing the following command depending on your architecture:

rpm -ivh gpfs.tct.server-x.x.x.x86_64.rpm

Note: By default, the Cloud services is installed under the /opt/ibm/MCStore directory.

3. Install the client package on the desired nodes by issuing the following command depending on your OS and architecture:

- For SLES or Red Hat Enterprise Linux: **rpm -ivh gpfs.tct.client-x.x.x.<arch>.rpm**
- For Ubuntu Linux: **dpkg -i gpfs.tct.client-x.x.x.<arch>.deb**

Setting up a Cloud services cluster

This topic provides step-by-step instructions for quickly setting up a Cloud services node class that can support Transparent cloud tiering or Cloud data sharing on the IBM Spectrum Scale cluster.

In the multi-node setup, a maximum of 4 of any combination of CES or NSD nodes are supported as a Cloud services node group -- a minimum of two are recommended for high availability. Data migration or recall workload is shared across all the nodes. During a transparent recall or migration, the ILM policy gathers a list of files to be migrated, and the list is divided equally among all the nodes.

Note: Once a multi-node setup is up and running, the Transparent cloud tiering GUI components are available on the IBM Spectrum Scale GUI.

The `mtime` attribute on the IBM Spectrum Scale file system indicates when a file is updated. For a Cloud services-managed file system, ensure that `mtime` parameter is set to "yes", so that when Cloud services internally invokes `stat()` or `fstat()` calls during migration, file system reports exact `mtime` value for a file. If `mtime` parameter is set to "no", there is a potential data loss situation, when a modified / updated co-resident file does not get migrated as a new version to the cloud. You can check the `mtime` status by using the **mmfsfs <file system> -E** command.

Perform the following steps for setting up a Cloud services cluster:

1. Install the Cloud services RPMs on the required IBM Spectrum Scale nodes. For more information, see [Chapter 7, "Installing Cloud services on IBM Spectrum Scale nodes,"](#) on page 475.
2. Designate the nodes where you installed the RPMs as Cloud services nodes. For more information, see the *Designating the transparent cloud tiering nodes* topic in the *IBM Spectrum Scale: Administration Guide*.
3. Start Cloud services on all nodes. For more information, see the *Starting the Cloud services* topic in the *IBM Spectrum Scale: Administration Guide*.
4. Create a cloud account. For simpler configurations, use the built-in key manager that comes with Cloud services. Use KLM when KLM is already running on Spectrum Scale (for other reasons) or if specific security requirements that drive its usage. For more information, see the *Creating a cloud storage account* topic in the *IBM Spectrum Scale: Administration Guide*.
5. Define cloud storage access points (CSAP). For more information, see the *Defining cloud storage access points (CSAP)* topic in the *IBM Spectrum Scale: Administration Guide*.
6. Create a Cloud services. For more information, see the *Creating a Cloud services* topic in the *IBM Spectrum Scale: Administration Guide*.
7. Configure a key manager. For more information, see the *Creating a key manager* topic in the *IBM Spectrum Scale: Administration Guide*.
8. Create a container pair set and test your configuration. For more information, see the *Creating a container pair set* topic in the *IBM Spectrum Scale: Administration Guide*.
9. Make sure you back up your configuration and your key manager. For more information, see the *Backing up the Cloud services configuration* topic in the *IBM Spectrum Scale: Administration Guide*.

If you are doing Transparent cloud tiering, do the following steps to verify that the cluster is ready for transparent recall/migration based on an ILM policy:

- Create a sample policy, *migrate.policy*, and apply this policy by using the Cloud services nodes by with the command: **mmapplypolicy gpfs -P migrate.policy -N FirstCloud**

Note: If you administer the `defaultHelperNodes` settings, you do not need to specify the `-N` parameter to distribute workload here.

You can confirm that data migration is happening across multiple nodes by looking at the following events:

- Each Transparent cloud tiering node has policy files under /tmp.
- Each Transparent cloud tiering node starts reflecting the migration metrics for the amount of data that is passed through it.

Cloud services run maintenance automatically in maintenance windows in the background. Do not forget to review the maintenance windows and adjust them appropriately. For more information, see the *Setting up maintenance tasks* topic in the *IBM Spectrum Scale: Administration Guide*.

Note: If you change the IP address or network settings of any of the Cloud services nodes once it is operational, do the following steps:

1. Stop the Cloud services (by using the **mmcloudgateway service stop** command), the way it is mandated to stop IBM Spectrum Scale services while you change the node IP address.
2. To change the IP address, it is recommended to follow the steps that are provided in the *Changing IP addresses and host names* topic in the *IBM Spectrum Scale: Administration Guide*.

For more information, see *Configuring and tuning your system for Cloud services* topic in the *IBM Spectrum Scale: Administration Guide*.

Adding a Cloud services node to an existing Cloud services cluster

This topic describes the procedure for adding a Cloud services client or a server to an existing Cloud services cluster.

1. Add the new node to the existing node class by using the **mmchnodeclass add** command. For example,

```
mmchnodeclass TCTNodeClass1 add -N node5
```

2. Install the Cloud services RPMs (either client or server) on the node that is added. For more information, see [“Installation steps” on page 476](#).
3. Designate the node as Cloud services node.

```
mmchnode --cloud-gateway-enable -N node5 --cloud-gateway-nodeclass TCTNodeClass1
```

For more information, see the *Designating the Cloud services nodes* topic in the *IBM Spectrum Scale: Administration Guide*.

4. Start the Cloud services software on the new node by using the **mmcloudgateway service start** command.

```
mmcloudgateway service start -N node5
```

Chapter 8. Installing and configuring IBM Spectrum Scale management API

The IBM Spectrum Scale management API implementation is based on the IBM Spectrum Scale GUI stack. Installation and configuration of GUI sets up the API infrastructure. That is, you do not need to perform any API-specific deployment procedures.

Ensure that the following tasks are completed to start using the IBM Spectrum Scale management API to monitor and manage the IBM Spectrum Scale system:

1. Install the GUI server. Use any of the following options to install the GUI:
 - a. *Manually installing IBM Spectrum Scale management GUI in IBM Spectrum Scale: Concepts, Planning, and Installation Guide*
 - b. *Installing IBM Spectrum Scale management GUI by using the spectrumscale installation toolkit in IBM Spectrum Scale: Concepts, Planning, and Installation Guide*
2. 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.
3. Create new users and provide the required permissions. You can either use the **Access > GUI Users** GUI page or the `usr/lpp/mmfs/gui/cli/mkuser` command to create new users and assign roles. For more information on how to create GUI users and assigning user roles, see *Managing GUI users* section in the *IBM Spectrum Scale: Administration Guide*.

Every API call to the IBM Spectrum Scale needs a valid GUI user and password. For example, `curl -k -u admin:admin001 -XGET -H content-type:application/json "https://<IP address>:443/scalemgmt/v2/info"`. The user roles assigned to the user determine actions that can be performed.

Chapter 9. Installing GPUDirect Storage for IBM Spectrum Scale

IBM Spectrum Scale support for GPUDirect Storage (GDS) enables a direct path between GPU memory and storage.

For information about the prerequisites, see [“Planning for GPUDirect Storage” on page 250](#).

Perform the following steps to install GDS:

1. Install IBM Spectrum Scale with NSD servers for your file system. For more information, see [Chapter 4, “Installing IBM Spectrum Scale on Linux nodes and deploying protocols,” on page 351](#).
2. Set up the InfiniBand fabric (or RoCE fabric by analogy). For details about the installation instructions, see [Mellanox OFED installation](#). Complete the following steps to configure InfiniBand fabric:
 - a) Use the latest MOFED driver. For details about the supported driver versions, see [Components required for GDS in IBM Spectrum Scale FAQ in IBM Documentation](#).
 - b) Set up IP over InfiniBand. For more information, see [Configuring InfiniBand and RDMA networks](#).
 - c) If your NSD servers are part of ESS, then reinstall the MOFED driver with the **--upstream-libs** flag. Use the MOFED version that ships with the ESS.
3. Install CUDA. For more information about the supported version of CUDA, see [Components required for GDS in IBM Spectrum Scale FAQ in IBM Documentation](#).

For CUDA installation details, see [NVIDIA GPUDirect Storage Installation and Troubleshooting Guide](#).

4. Configure IBM Spectrum Scale. For more information, see [Configuring the GPFS cluster in IBM Spectrum Scale: Administration Guide](#).
5. Configure the NVIDIA components. For more information, see [Configuring GDS in IBM Spectrum Scale: Administration Guide](#).
6. Start IBM Spectrum Scale by using the **mmstartup -a** command.

Checking the installation

Perform the following steps to check the installation of GDS:

1. Run the NVIDIA GDS utility **gdscheck -p** before you run the GDS workloads to verify the environment. You need Python3 installed on the node to run this utility.
2. Verify the status of PCIe Access Control Services (ACS) and PCIe Input/Output Memory Management Unit (IOMMU), as these components affect GDS function and performance. The output of the **gdscheck -p** utility must display the following status for IOMMU and ACS components: :

```
IOMMU disabled
ACS disabled
```

3. Check for IBM Spectrum Scale support in the output of **gdscheck -p** as shown in the following example:

```
# gdscheck -p | grep "Spectrum Scale"
IBM Spectrum Scale : Supported
```

4. If the GPFS log (mmfs log) contains the following information after you start IBM Spectrum Scale, it indicates that the GDS support is successfully enabled:

```
[I] VERBS DC API loaded.
```

and

```
[I] VERBS DC API initialized.
```

Related concepts

[“GPUDirect Storage support for IBM Spectrum Scale” on page 26](#)

IBM Spectrum Scale's support for NVIDIA's GPUDirect Storage (GDS) enables a direct path between GPU memory and storage. This solution addresses the need for higher throughput and lower latencies. File system storage is directly connected to the GPU buffers to reduce latency and load on CPU. For IBM Spectrum Scale, this means that data can be read directly from an NSD server's pagepool and it is sent to the GPU buffer of the IBM Spectrum Scale clients by using RDMA. IBM Spectrum Scale with GDS requires an InfiniBand or RoCE fabric. In IBM Spectrum Scale, the **mmdia** command is enhanced to print diagnostic information for GPUDirect Storage.

[“Planning for GPUDirect Storage” on page 250](#)

IBM Spectrum Scale support for GPUDirect Storage (GDS) enables a direct path between GPU memory and storage. You need to ensure that certain conditions are met before you start installing the feature.

Related tasks

[“Upgrading GPUDirect Storage” on page 517](#)

You need to upgrade your IBM Spectrum Scale cluster to 5.1.2 or later to start using the GPUDirect Storage (GDS).

Chapter 10. Installation of Active File Management (AFM)

Consider the following while you install Active File Management (AFM).

The home and cache are two separate operational clusters that are separated from each other through LAN or WAN. The cluster setup has no considerations for using AFM functions. AFM functions are available on all editions of IBM Spectrum Scale.

The home cluster can be a legacy NAS or a cluster that is running GPFS version 3.5 or earlier, or IBM Spectrum Scale 4.1 or later. The cache cluster can run GPFS 3.5, or IBM Spectrum Scale 4.1 or later. User-extended attributes, ACLs, and sparse files are not supported on a home cluster that is running GPFS version 3.5 or earlier, if the home cluster is a legacy NFS export.

The NFS version 3 and 4 or GPFS protocol can be used for communication. The native NSD protocol can be used when the home cluster is also an IBM Spectrum Scale cluster and the cache can remotely mount the file system on the home cluster.

Nodes that can act as gateway nodes must be identified on the cache cluster. Gateway nodes must preferably be configured in the cache cluster before you create filesets and start applications.

Nodes that can act as NFS servers must be identified on the home cluster. If the GPFS protocol is planned to be used, the home file system must be remote mounted on all the gateway nodes in the cache cluster. Gateway nodes and NFS servers must preferably be configured in the cache and home clusters before you create filesets and start applications.

AIX AFM gateway nodes in the AFM cache cluster must be running the Linux operating system.. AIX cannot be configured either as a home or AFM-DR secondary.

Chapter 11. Installing AFM Disaster Recovery

The primary and the secondary are two separate operational clusters that are separated from each other through LAN or WAN. AFM-DR functions are available in Advanced Edition, Data Management Edition, Developer Edition, or IBM Spectrum Scale Erasure Code Edition. All nodes in both the primary and secondary clusters must be running the Advanced Edition, Data Management Edition, Developer Edition, or Erasure Code Edition.

The NFS version 3 and 4 or native NSD protocol can be used for communication. Similar to AFM filesets, gateway nodes and NFS servers must be planned on the primary and secondary clusters. Gateway nodes and NFS servers must preferably be configured in the primary and secondary clusters before you create filesets and start applications. If you are planning to use native NSD protocol, the primary file system must be remote mounted on all the gateway nodes in the secondary cluster.

Note: While you upgrade AFM-DR to IBM Spectrum Scale 4.2.2 or later, the primary cluster must be upgraded before you upgrade the secondary cluster.

Chapter 12. Installing call home

The call home feature collects files, logs, traces, and details of certain system health events from different nodes and services. These details are shared with the IBM support center for monitoring and problem determination.

Prerequisites for installing call home

The following criteria must be met to use the call home function on your GPFS cluster:

- Red Hat Enterprise Linux 7.x, Red Hat Enterprise Linux 8.x, Ubuntu 20.04 LTS, and SLES 15 nodes are supported.
- Intel x86_64, PowerPC LE/BE and IBM Z are supported.
- GPFS CCR (Clustered Configuration Repository) must be enabled.
- The call home node must be able to access the following IP addresses and ports. In the configurations with a proxy, these IP addresses and ports must be accessible over the proxy connection.
 - Host name: `esupport.ibm.com`
 - IP address: `129.42.56.189`, `129.42.60.189`, `129.42.54.189`
 - Port number: `443`

The recommendation is to open `129.42.0.0/18`.

`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.

- The call home node needs to have at least 1 GB, preferably 3 GB of free space in the `dataStructureDump` directory. The current `dataStructureDump` directory can be determined by using the following command:

```
mmdiag --config | grep "dataStructureDump "
```

The current `dataStructureDump` directory is `tmp/mmfs` by default. You can change the default value of the `dataStructureDump` directory by using the **`mmchconfig`** command.

- There must not be multiple nodes with the same short host name within the GPFS cluster.
- In the case of the customer-provided remote shell or remote file-copy commands, the customer must make sure that these commands support the following options:

```
'-p -o BatchMode=yes -o ConnectTimeout=3'
```

Note: `ssh` and `scp` support these options.

The configured commands can be seen by running the following command:

```
[root@g5050-11 ~]# mmlscluster | grep Remote
Remote shell command: /usr/bin/ssh
Remote file copy command: /usr/bin/scp
```

- When you are using the `sudo` wrappers and a custom `dataStructureDump` directory, you must ensure that this directory is recursively executable for the `sudo` user, as specified in the IBM Spectrum Scale settings.

Note: In this case, the `dataStructureDump` directory is not set to the default value of `/tmp/mmfs`.

Steps for installing call home

The call home functionality is included in the `gpfs.base` package. If the IBM Spectrum Scale packages are installed and the cluster is created, no other steps are required for installing call home.

The installation toolkit can be used for removing call home RPMs from the old IBM Spectrum Scale versions, and for enabling and configuring the call home feature. For more information, see [“Enabling and configuring call home using the installation toolkit”](#) on page 424.

For information on configuring call home manually, see the *Monitoring the IBM Spectrum Scale system by using call home* section in the *IBM Spectrum Scale: Problem Determination Guide*.

Chapter 13. Installing file audit logging

Before file audit logging can be installed, consider the requirements, limitations, and steps associated with it.

For more information about enabling file audit logging, see *Configuring file audit logging* in the *IBM Spectrum Scale: Administration Guide*.

Requirements, limitations, and support for file audit logging

Use this information to understand requirements, limitations, and support for installing file audit logging.

OS requirements

File audit logging is supported on all Linux OS versions supported by IBM Spectrum Scale. For more information, refer to <https://www.ibm.com/docs/en/spectrum-scale?topic=STXKQY/gpfsclustersfaq.html>.

Security requirements and limitations

- Root authority is required to run **mmaudit**.

Restrictions imposed by mixed environments and protocols

- Events generated on non-Linux nodes are not audited.
- IBM Spectrum Scale file audit logging has full support for the following protocols (support for all other protocols must be considered limited):
 - NFS ganesha
 - SMB
 - Native UNIX file access
- Events are not generated at or below the `cesSharedRoot` path.

File audit logging attributes availability and limitations

- For more information about the availability and limitations of the file audit logging attributes, see the [“JSON attributes in file audit logging” on page 181](#) topic.

GPFS file system requirements and limitations

- File audit logging can be enabled only for file systems that are created or upgraded to IBM Spectrum Scale 5.0.0 or later.
- Space provisioning must be considered to store the generated events in the `.audit_log` fileset.
- The `.audit_log` fileset is protected from tampering. It cannot be easily deleted to free up space in the file system. This is done by creating the fileset in the IAM noncompliant mode (default) or compliant mode, which allows expiration dates to be set on the files containing the audit records within the fileset. If the fileset is created in IAM mode noncompliant, then the root user can change the expiration date to the current date so that audit files can be removed to free up disk space. If the fileset is created in IAM mode compliant (because of the use of the **--compliant** option), not even the root user can change the expiration dates of the audit logging files and they cannot be removed until the expiration date.
- Events are not generated for file system activity within the file audit logging fileset itself.
- There is a limit of 20 filesets for file system prior to 5.1.3 (27.0 file system version).

GPFS and spectrumscale functional limitations

- Conversion of a file audit logging fileset to AFM DR is not supported.
- When file audit logging or clustered watch folder is enabled on a file system, changing the file system name or deleting the file system is not allowed. To change file system name or delete the file system, file audit logging and clustered watch folder must first be disabled.

Miscellaneous requirements, limitations, recommendations, and support statements

- File audit logging is available in IBM Spectrum Scale Advanced Edition, IBM Spectrum Scale Data Management Edition, IBM Spectrum Scale Developer Edition, or IBM Spectrum Scale Erasure Code Edition.
- File audit logging is supported in SELinux enforcing, permissive, and disabled modes. When file audit logging is run in enforcing mode, there is an extra event generated that is related to attributes due to the SELinux labeling of files.
- File audit logging uses buffered IO. This means that when a file system operation generates an event, the event is then appended to a buffer rather than writing directly to the disk. Therefore, if a file system operation returns successfully, it is not guaranteed that the operation would be logged in the audit log. For example, if the node goes down after an operation completes, but before the buffered IO is written to disk, then that operation will never be written to the audit log.
- File audit logging is not supported by Kafka message queue in IBM Spectrum Scale 5.1.1 or later.
- The **mmrestorefs** command does not restore files that are located in the `.audit_log` fileset or configuration fileset. The current configuration would not be overwritten and audit records would not be removed or restored.

Requirements for using file audit logging with remotely mounted file systems

Use this information to determine the requirements for using file audit logging with remotely mounted file systems.

All clusters that own or access file systems that are being audited by file audit logging through remote mounts must meet the following requirements:

- All clusters must be at IBM Spectrum Scale 5.0.2 or later.
- For any cluster that is upgraded to IBM Spectrum Scale 5.0.2 or later, the **mmchconfig release=LATEST** command must be run.
- Any file system that is audited must be at IBM Spectrum Scale 5.0.2 or later. If upgraded, the **mmchfs -V full** command must be run after the cluster is installed at or upgraded to IBM Spectrum Scale 5.0.2 or later.
- The **mmaudit** command can be run only on the cluster that owns the file system with file audit logging enabled.
- Clusters that are remotely mounting file systems with file audit logging enabled must be upgraded before the owning cluster.

Chapter 14. Installing clustered watch folder

Before clustered watch folder can be installed, consider the requirements, limitations, and steps associated with it.

For more information, see [“Clustered watch folder” on page 184](#).

Requirements, limitations, and support for clustered watch folder

Use this information to understand the requirements, limitations, and support for installing clustered watch folder.

RPM and package requirements

In order for watch folder events to be generated from a node, the node must have the following packages installed:

- For RHEL and SLES, the `librdkafka` package requires the `openssl-devel` and `cyrus-sasl-devel` packages.
- For Ubuntu, the `librdkafka` package requires the `libssl-dev` and `libsasl2-dev` packages.
- `librdkafka` (`gpfs.librdkafka rpm/package`).

OS and hardware requirements

Note: Not all of these requirements apply to the support of remotely mounted file systems.

- RHEL greater or equal to 7.7 (x86, PPC64LE, and s390x).
- RHEL 8 (x86, PPC64LE, and s390x).
- SLES 15 (x86 and s390x)
- Ubuntu 20.04 (x86)
- Linux Kernel on all platforms must be greater than or equal to RHEL 7.0 3.10.0-123.

Security requirements and limitations

- Root authority is required to run `mmwatch`.

Restrictions imposed by mixed environments and protocols

- Events will not be generated on non-Linux nodes.
- IBM Spectrum Scale clustered watch folder has full support for the following protocols (support for all other protocols must be considered limited):
 - NFS-Ganesha
 - SMB
 - Native UNIX file access

Clustered watch folder attributes

For more information about the availability and limitations of the clustered watch folder attributes, see [“JSON attributes in clustered watch folder” on page 186](#).

GPFS file system requirements and limitations

- Clustered watch folder can be enabled only for file systems that are created or upgraded to IBM Spectrum Scale 5.0.3 or later.
- When clustered watch folder is enabled on a file system, changing the file system name is not allowed. Clustered watch folder must first be disabled to change the file system name.

Miscellaneous requirements, limitations, recommendations, and support statements

- Clustered watch folder is available in IBM Spectrum Scale Advanced Edition, IBM Spectrum Scale Data Management Edition, IBM Spectrum Scale Developer Edition, or IBM Spectrum Scale Erasure Code Edition.
- Only 25 cluster watches are supported per cluster.

Note:

- For more information about setting up a basic external sink to push events to, see <https://kafka.apache.org/quickstart>.
- If you want to use a specific network interface for the brokers in the external Kafka sink, specify the **advertised.listeners** configuration parameter for the brokers with the IP address on the specified network interface card.
- Clustered watch folder is supported in SELinux enforcing, permissive, and disabled modes. When clustered watch folder is run in enforcing mode, there is an extra event generated that is related to attributes due to the SELinux labeling of files.

Requirements for using clustered watch folder with remotely mounted file systems

These requirements must be followed to use clustered watch folder with remotely mounted file systems.

All clusters that own or access file systems that are using clustered watch folder through remote mounts must meet the following requirements:

- All clusters must be at IBM Spectrum Scale 5.0.3 or later.
- For any cluster that is upgraded to IBM Spectrum Scale 5.0.3 or later, the **mmchconfig release=LATEST** command must be run.
- Any file system that is watched must be at IBM Spectrum Scale 5.0.3 or later. If upgraded, the **mmchfs -V full** command must be run after the cluster is installed at or upgraded to IBM Spectrum Scale 5.0.3 or later.
- The **mmwatch** command can be run on the cluster that owns the file system that is using clustered watch folder only.
- Clusters that are remotely mounting file systems that are using clustered watch folder must be upgraded before the owning cluster.

Manually installing clustered watch folder

Use this information to manually install clustered watch folder on your system.

Important: Before you can install clustered watch folder, you must have the required packages for IBM Spectrum Scale installed on the node.

Ensure that the following software package is installed on every node in the cluster that acts as producer.

- For Red Hat Enterprise Linux:
 - `gpfs.librdkafka-5.1*.rpm`

Note: This package is available in `/usr/lpp/mmfs/5.1.x.x/gpfs_rpms/rhelX`.

- For Ubuntu Linux:
 - `gpfs.librdkafka-5.1*.deb`

Note: This package is available in `/usr/lpp/mmfs/5.1.x.x/gpfs_debs/ubuntu`.

Note: For more information, see [“Requirements, limitations, and support for clustered watch folder” on page 491](#).

1. If you are installing the package for clustered watch folder on existing IBM Spectrum Scale nodes and this is not part of a cluster installation or upgrade, you must shut down and then restart the daemon in order for the producer to be able to load the required library.
2. After the IBM Spectrum Scale cluster is available and the file systems are mounted, use the **mmwatch** command for each file system that you want to enable for clustered watch folder.

Note: For more information about enabling clustered watch folder, see the *mmwatch command* in the *IBM Spectrum Scale: Command and Programming Reference*.

Chapter 15. Steps to permanently uninstall IBM Spectrum Scale

IBM Spectrum Scale maintains a number of files that contain configuration and file system related data. Since these files are critical for the proper functioning of IBM Spectrum Scale and must be preserved across releases, they are not automatically removed when you uninstall GPFS.

Follow these steps if you do not intend to use IBM Spectrum Scale on any of the nodes in your cluster and you want to remove all traces of IBM Spectrum Scale:

Attention: After following these steps and manually removing the configuration and file system related information, you will permanently lose access to all of your current IBM Spectrum Scale data.

1. Unmount all GPFS file systems on all nodes by issuing the `mmumount -a` command.
2. Issue the `mmde1fs` command for each file system in the cluster to remove GPFS file systems.
3. Issue the `mmde1nsd` command for each NSD in the cluster to remove the NSD volume ID from the device.

If the NSD volume ID is not removed and the disk is again used with GPFS at a later time, you receive an error message when you specify the `mmcrnsd` command. For more information, see *NSD creation fails with a message referring to an existing NSD* in *IBM Spectrum Scale: Problem Determination Guide*.

4. Issue the `mmshutdown -a` command to shutdown GPFS on all nodes.
5. Uninstall IBM Spectrum Scale from each node:
 - For your Linux nodes, run the uninstallation program to remove GPFS for the correct version of the packages for your hardware platform and Linux distribution. For example, on SLES and Red Hat Enterprise Linux nodes:

```
rpm -e gpfs.crypto (IBM Spectrum Scale Advanced Edition, IBM Spectrum Scale Data
Management Edition, or IBM Spectrum Scale Developer Edition only)
rpm -e gpfs.adv (IBM Spectrum Scale Advanced Edition, IBM Spectrum Scale Data
Management Edition, or IBM Spectrum Scale Developer Edition only)
rpm -e gpfs.gpl
rpm -e gpfs.license.xx
rpm -e gpfs.msg.en_US
rpm -e gpfs.compression
rpm -e gpfs.base
rpm -e gpfs.docs
rpm -e gpfs.gskit
```

On Ubuntu Linux nodes:

```
dpkg -P gpfs.crypto (IBM Spectrum Scale Advanced Edition and IBM Spectrum Scale Data
Management Edition only)
dpkg -P gpfs.adv (IBM Spectrum Scale Advanced Edition and IBM Spectrum Scale Data
Management Edition only)
dpkg -P gpfs.gpl
dpkg -P gpfs.license.xx
dpkg -P gpfs.msg.en-us
dpkg -P gpfs.compression
dpkg -P gpfs.base
dpkg -P gpfs.docs
dpkg -P gpfs.gskit
```

- For your AIX nodes:

```
installp -u gpfs
```

- For your Windows nodes, follow these steps:
 - a. Open Programs and Features in the Control Panel.
 - b. Uninstall IBM General Parallel File System.
 - c. Reboot the system.
 - d. From a Command Prompt, run the following command:

```
sc.exe delete mmwinserv
```
- 6. Remove the `/var/mmfs` and `/usr/lpp/mmfs` directories.
- 7. Remove all files that start with `mm` from the `/var/adm/ras` directory.
- 8. Remove `/tmp/mmfs` directory and its content, if present.

Cleanup procedures required if reinstalling with the installation toolkit

Before you can reinstall with the installation toolkit, you must perform some cleanup procedures first.

You can use the following procedures to clean up various stages of the previous GPFS installation and protocols deployment.

Starting the cleanup process

1. Clean up the installer directory by issuing the following command:

```
mmdsh -N NodeList rm -rf /usr/lpp/mmfs/5.1.5.x
```

Note: The installer directory in this command depends upon the release version.

2. Check `/etc/resolv.conf` for any changes that pointed the authentication directory to DNS, and remove them.
3. If the object protocol was enabled and storage policies were created, use the **mmobj policy list -v** command and save the output list before you run the **mmces service disable OBJ** command.

Uninstalling Ansible

The installation toolkit is built on the Ansible automation platform. For more information, see [“Supported Ansible version”](#) on page 405.

You might need to uninstall Ansible as a part of the cleanup process or if the supported version is not installed. Check the Ansible version that is installed and, if needed, uninstall Ansible after verifying that it is not being used for any other purpose.

1. Check the version of Ansible that is installed by using one of the following commands.

```
ansible --version
```

or

```
rpm -qa | grep ansible
```

2. Uninstall Ansible by using one of the following commands depending on your environment.

- Uninstall by using the **rpm** command on RHEL and SLES nodes.

```
rpm -qa | grep ansible | xargs rpm -e
```


- Uninstall by using the **yum** command on RHEL nodes.

```
yum remove ansible
```

- Uninstall by using the **pip3** command on RHEL, SLES, and Ubuntu nodes.

```
pip3 uninstall ansible
```

- Uninstall by using the **apt-get** command on Ubuntu nodes.

```
apt-get purge ansible
```

Cleaning up the authentication configurations

Use these commands to completely remove the current authentication configuration for all protocols, NFS/SMB only, or Object only. Be aware that doing so may cause loss of access to previously written data.

1. Remove the configurations. To do so, issue the following commands:

```
mmuserauth service remove --data-access-method file
mmuserauth service remove --data-access-method object
```

2. Delete the ID-mapping information when authentication was configured with AD. To do so, use the following command.

Note: Once this ID-mapping information is deleted, you might not be able to access existing files (or you might experience other unforeseen access issues), so do this only if you do not need access to existing data.

```
mmuserauth service remove --data-access-method file --idmapdelete
```

Disabling CES

Disable CES on every node *nodeNameX*. To do so, issue the following command:

```
mmchnode -N nodeNameX --ces-disable
```

Protocol cleanup steps

1. Use the following steps to clean up some of the object state on the system:

```
mmdsh -N cesNodes systemctl stop postgresql-obj
rm -rf path_to_cesSharedRoot/object/keystone
```

Delete the objectization temp directory. In the following example, the temp directory was created in the fs1 file system at /ibm/fs1/ibmobjectizer/tmp:

```
rm -rf /ibm/fs1/ibmobjectizer/tmp
```

2. Remove the fileset created for object.

In the following example, the configured fileset *object_fileset* in file system *fs1* was linked at /gpfs/fs1.

```
mmfilesfileset fs1
mmunlinkfileset fs1 object_fileset
mmdelfileset fs1 object_fileset -f
```

3. Remove any fileset created for an object storage policy.

Run the **mmobj policy list -v** command. If for some reason **mmobj policy list -v** cannot be run (For example, if the Object Protocol or CES was already disabled) or you cannot detect which filesets are used for object storage policies, contact the IBM Support Center.

For example:

If the **mmobj policy list** command returned the following and the filesets got created in file system fs1:

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	mysofpolicy	obj_mysofpolicy	/ibm/cesSharedRoot/obj_mysofpolicy	file-and-object-access	regions="1"
11751510260	Test19	obj_Test19	/ibm/cesSharedRoot/obj_Test19		regions="1"

Use the following commands for each row except the first:

```
mmunlinkfileset fs1 sof-policy1
mmdelfileset fs1 sof-policy1 -f
```

4. Clear the cesSharedRoot configuration by issuing the following command:

```
mmchconfig cesSharedRoot=DEFAULT
```

5. Check your recent package installs to determine which packages need to be removed by issuing the following commands:

RHEL and SLES

```
rpm -qa --last|more
```

Ubuntu

```
dpkg -l
```

After you have determined which packages need to be removed, do so by issuing the following commands, as needed:

RHEL and SLES

```
mmdsh -N NodeList yum erase gpfs.smb nfs-ganesha-mount \
nfs-ganesha-vfs PyQt4 nfs-ganesha-utils phonon-backend-gstreamer phonon sip qt-x11 kde-fileSYSTEM \
qt qt-settings libmng nfs-ganesha-proxy nfs-ganesha-nullfs nfs-ganesha-gpfs -y

mmdsh -N NodeList yum erase python-webob MySQL-python mariadb-server perl-DBD-MySQL mariadb \
python-futures python-warlock python-jsonpatch python-jsonschema python-jsonpointer python-cmd2 \
python-cliff pyparsing memcached python-oslo-utils python-oslo-serialization python-oslo-i18n \
python-babel babel python-keyring python-stevedore python-pytablewriter python-pbr python-oslo-config \
python-netaddr python-iso8601 python-dnspython python-urllib3 python-six python-requests \
python-paste-deploy python-tempita python-paste python-netifaces python-simplejson \
python-greenlet python-eventlet -y

mmdsh -N NodeList yum erase python2-keystoneauth1 python2-keystoneclient python2-keystonemiddleware \
python2-openstackclient python2-oslo-cache python2-oslo-concurrency python2-oslo-config \
python2-oslo-context python2-oslo-db python2-oslo-i18n python2-oslo-middleware python2-oslo-policy \
python2-oslo-service python2-swiftclient python3-keystone python3-keystoneauth1 python3-keystoneclient \
python3-keystonemiddleware python3-openstackclient python3-openstacksdk python3-oslo-cache \
python3-oslo-concurrency python3-oslo-config python3-oslo-context python3-oslo-db python3-oslo-i18n \
python3-oslo-log python3-oslo-messaging python3-oslo-middleware python3-oslo-policy \
python3-oslo-serialization python3-oslo-service python3-oslo-upgradecheck python3-oslo-utils \
python3-swift python3-swiftclient python-openstackclient-lang python-openstacksdk \
python-oslo-cache-lang python-oslo-concurrency-lang python-oslo-db-lang python-oslo-i18n-lang \
python-oslo-log python-oslo-log-lang python-oslo-messaging python-oslo-middleware-lang \
python-oslo-policy-lang python-oslo-utils-lang python-swift -y

mmdsh -N NodeList yum erase gpfs.smb-debugsource -y

mmdsh -N NodeList yum erase sssd-tools krb5-workstation

mmdsh -N NodeList yum erase python-routes python-repoze-lru python-oauthlib python-crypto \
python-qpid-common python-qpid python-dogpile-cache python-fixtures python-dogpile-core \
python-testtools python-extras python-oslo-context python-kombu python-anyjson python-amqp \
python-mimeparse python-markupsafe PyYAML python-passlib libyaml python-ibm-db-sa python-ibm-db \
python-sqlparse python-memcached python-posix_ipc python-sqlalchemy -y

mmdsh -N NodeList yum erase gpfs.gss.pmsensors gpfs.gss.pmcollector gpfs.pm-ganesha boost-regex -y

mmdsh -N NodeList yum erase openstack-utils -y

mmdsh -N NodeList yum erase python-ordereddict -y
```

```

mmdsh -N NodeList yum erase mariadb-libs -y

mmdsh -N NodeList yum erase pmswift -y

mmdsh -N NodeList yum erase python-cryptography python-enum34 swift3 python-pyeclib liberasurecode \
openstack-utils crudini python-msgpack python-wsgiref python-pycparser python-ply python-cffi \
openstack-selinux xmlsec1 python-pyasn1 python-retrying \
postgresql-server postgresql python-psycpg2 python-repoze-who jerasure gf-complete \
python-zope-interface postgresql-libs pytz -y

mmdsh -N NodeList yum erase spectrum-scale-object spectrum-scale-object-selinux \
openstack-swift openstack-swift-account openstack-swift-container openstack-swift-object \
openstack-swift-proxy python-swiftclient swiftonfile keystonemiddleware openstack-keystone \
python-keystone python-cinderclient python-glanceclient python-keystoneclient python-neutronclient \
python-novaclient python-keystoneauth1 python-openstackclient -y

```

Ubuntu

```

mmdsh -N NodeList dpkg -P gpfs.smb nfs-ganesha-mount nfs-ganesha-vfs \
PyQt4 nfs-ganesha-utils phonon-backend-gstreamer phonon sip qt-x11 kde-filesystem qt qt-settings libmng \
nfs-ganesha-proxy nfs-ganesha-nullfs gpfs.nfs-ganesha-gpfs gpfs.nfs-ganesha -y

mmdsh -N NodeList dpkg -P python-webob MySQL-python mariadb-server perl-DBD-MySQL mariadb \
python-futures python-warlock python-jsonpatch python-jsonschema python-jsonpointer python-cmd2 \
python-cliff pyparsing memcached python-oslo-utils python-oslo-serialization python-oslo-i18n \
python-babel babel python-keyring python-stevedore python-pytable python-pbr python-oslo-config \
python-netaddr python-iso8601 python-dnspython python-uulib3 python-six python-requests \
python-paste-deploy python-tempita python-paste python-netifaces python-simplejson \
python-greenlet python-eventlet -y

mmdsh -N NodeList dpkg -P python-routes python-repoze-lru python-oauthlib python-crypto \
python-qpid-common python-qpid python-dogpile-cache python-fixtures python-dogpile-core \
python-testtools python-extras python-oslo-context python-kombu python-anyjson python-amqp \
python-mimeparse python-markupsafe PyYAML python-passlib libyaml python-ibm-db-sa python-ibm-db \
python-sqlparse python-memcached python-posix_ipc python-sqlalchemy -y

mmdsh -N NodeList dpkg -P gpfs.gss.pmsensors gpfs.gss.pmcollector gpfs.pm-ganesha boost-regex -y

mmdsh -N NodeList dpkg -P openstack-utils gpfs.gpfs.nfs-ganesha-gpfs -y

mmdsh -N NodeList dpkg -P python-ordereddict -y

mmdsh -N NodeList dpkg -P mariadb-libs -y

mmdsh -N NodeList dpkg -P pmswift -y

mmdsh -N NodeList dpkg -P python-cryptography python-enum34 swift3 python-pyeclib liberasurecode \
crudini python-msgpack python-wsgiref python-pycparser python-ply python-cffi \
openstack-selinux xmlsec1 python-pyasn1 python-retrying postgresql-server postgresql \
python-psycpg2 python-repoze-who jerasure gf-complete python-zope-interface postgresql-libs \
pytz -y

mmdsh -N NodeList dpkg -P spectrum-scale-object spectrum-scale-object-selinux \
openstack-swift openstack-swift-account openstack-swift-container openstack-swift-object \
openstack-swift-proxy python-swiftclient swiftonfile keystonemiddleware openstack-keystone \
python-keystone python-keystoneauth1 python-cinderclient python-glanceclient python-keystoneclient \
python-neutronclient python-novaclient python-openstackclient -y

```

File protocols authentication packages cleanup

RHEL

```

mmdsh -N NodeList yum erase bind-utils krb5-workstation openldap-clients sssd-tools ypbind yp-tools -y

```

SLES

```

mmdsh -N NodeList yum erase bind-utils krb5-client openldap2-client sssd-tools ypbind yp-tools -y

```

Ubuntu

```

mmdsh -N NodeList dpkg -P dnsutils krb5-user ldap-utils sssd-tools nis libslp1 -y

```

Note: Plan to erase the SSSD packages only if you do not plan to use the SSSD service on the nodes for any other purpose.

SSSD packages cleanup commands are as follows:

RHEL

```

mmdsh -N NodeList yum erase sssd*

```

SLES

```
mmdsh -N NodeList yum erase sssd*
```

Ubuntu

```
mmdsh -N NodeList dpkg -P sssd*
```

6. Issue **rpm -qa --last|more** or **dpkg -l** on all nodes again, as the preceding list does not contain everything in the new builds.

If the pmswift package still exists, remove it as follows:

RHEL and SLES

```
mmdsh -N NodeList rpm -e pmswift-5.1.5-x.noarch --noscripts
```

Ubuntu

```
mmdsh -N NodeList dpkg -P pmswift-5.1.5-x.noarch --noscripts
```

You might also need to remove `gpfs.smb`.

7. Clean up performance monitoring tool files. To do so, issue the following commands:

```
mmdsh -N NodeList rm -rf /opt/IBM/zimon*
mmdsh -N NodeList rm -rf /usr/IBM/zimon*
mmdsh -N NodeList rm -rf /var/log/cnlog/zimon*
mmdsh -N NodeList rm -rf /var/lib/yum/repore/x86_64/7Server/*zimon*
mmdsh -N NodeList rm -rf /var/lib/yum/repore/ppc64/7Server/*zimon*
```

8. Clean up CTDB. To do so, issue the following commands:

```
mmdsh -N NodeList rm -rf /var/lib/ctddb
```

9. Clean up Swift files. To do so, issue the following commands:

```
mmdsh -N NodeList rm -rf /etc/swift
mmdsh -N NodeList rm -rf /var/lib/mysql
mmdsh -N NodeList rm -rf /etc/keystone
mmdsh -N NodeList rm -rf /var/lib/keystone
mmdsh -N NodeList rm -rf /root/openrc
mmdsh -N NodeList rm -rf /var/cache/yum/x86_64/7Server/*
mmdsh -N NodeList rm -rf /var/cache/yum/ppc64/7Server/*
mmdsh -N NodeList rm -rf /var/log/maria*
mmdsh -N NodeList rm -rf /usr/share/pixmaps/comps/maria*
mmdsh -N NodeList rm -rf /var/log/keystone
mmdsh -N NodeList rm -rf /var/spool/cron/keystone
mmdsh -N NodeList rm -rf /usr/lib/python2.7/site-packages/sos/plugins/openstack_keystone*
mmdsh -N NodeList rm -rf /tmp/keystone-signing-swift
mmdsh -N NodeList rm -rf /usr/lib/python2.7/site-packages/sos/plugins/*
mmdsh -N NodeList rm -rf /var/lib/yum/repos/x86_64/7Server/icm_openstack
mmdsh -N NodeList rm -f /usr/lib/python2.7/site-packages/swiftonfile-2.5.0-py2.7.egg*
mmdsh -N NodeList rm -f /usr/bin/*objectizer*.pyc
mmdsh -N NodeList rm -f /usr/bin/generate_dbmap.pyc
mmdsh -N NodeList systemctl disable openstack-keystone.service
mmdsh -N NodeList systemctl disable openstack-swift-container-updater.service
mmdsh -N NodeList systemctl disable openstack-swift-container-update
mmdsh -N NodeList systemctl disable openstack-swift-object-updater.service
mmdsh -N NodeList systemctl disable openstack-swift-container-auditor.service
mmdsh -N NodeList systemctl disable openstack-swift-container-replicator.service
mmdsh -N NodeList systemctl disable openstack-swift-container.service
mmdsh -N NodeList systemctl disable openstack-swift-object-replicator.service
mmdsh -N NodeList systemctl disable openstack-swift-object.service
mmdsh -N NodeList systemctl disable openstack-keystone.service
mmdsh -N NodeList systemctl disable openstack-swift-account-reaper.service
mmdsh -N NodeList systemctl disable openstack-swift-account-auditor.service
mmdsh -N NodeList systemctl disable openstack-swift-account-replicator.service
mmdsh -N NodeList systemctl disable openstack-swift-account.service
mmdsh -N NodeList systemctl disable openstack-swift-proxy.service
mmdsh -N NodeList systemctl disable openstack-swift-object-auditor.service
mmdsh -N NodeList systemctl disable openstack-swift-object-expirer.service
mmdsh -N NodeList systemctl disable openstack-swift-container-reconciler.service
mmdsh -N NodeList rm -rf
/var/lib/yum/yumdb/o/0b01eb65826df92befd8c161798cb842fa3c941e-openstack-
utils-2015.1-201502031913.ibm.el7.7-noarch
```

10. Reboot the nodes. To do so, issue the following command:

```
mmdsh -N NodeList shutdown -r now
```

If some nodes do not fully come up, do the following:

- a. Power off that node.
- b. Wait for a minute, then power on the node back.

Note: Some nodes might need to be powered off and on more than once.

11. Clean up Yum on all nodes. To do so, issue the following commands:

```
mmdsh -N NodeList yum clean all
mmdsh -N NodeList rm -rf /etc/yum.repos.d/gpfs.repo /etc/yum.repos.d/icm* /etc/yum.repos.d/
ces.repo
mmdsh -N NodeList rm -rf /etc/yum.repos.d/epel* /etc/yum.repos.d/rdc*
mmdsh -N NodeList rm -rf /var/lib/yum/repos/x86_64/7Server/*
mmdsh -N NodeList rm -rf /var/lib/yum/repos/ppc64/7Server/*
```

12. Remove GPFS.

Notes:

- This step is not required if you know that GPFS has not changed between old and new builds.
- This step is not required if you prefer to perform an upgrade of GPFS.
- This step is not required if you would like to keep the base GPFS installation intact and merely rerun the protocols deployment.
- (To permanently remove GPFS, see [Chapter 15, “Steps to permanently uninstall IBM Spectrum Scale,”](#) on page 495.)

a. Check which GPFS packages are on each node by issuing the following command:

RHEL and SLES

```
mmdsh -N NodeList rpm -qa|grep gpfs
```

Ubuntu

```
mmdsh -N NodeList dpkg -l|grep gpfs
```

The system displays output similar to the following:

```
gpfs.msg.en_US-5.1.5-x.noarch
gpfs.gskit-8.0.55.x.x86_64
gpfs.license.xx-5.1.5-x.x86_64
gpfs.crypto-5.1.5-x.x86_64
gpfs.adv-5.1.5-x.x86_64
gpfs.docs-5.1.5-x.noarch
gpfs.base-5.1.5-x.x86_64
gpfs.gpl-5.1.5-x.noarch
```

b. Before removing anything, make sure that GPFS is shut down on all nodes. To do so, issue the following command:

```
mmshutdown -a
```

c. Remove the packages by issuing the following commands *in the order shown*.

Note: When you remove the `gpfs.base` package, you lose the `mmdsh` access.

Therefore, be sure to remove `gpfs.base` last, as shown here.

RHEL and SLES

```
mmdsh -N NodeList rpm -e gpfs.base-debuginfo-5.1.5-x.x86_64
mmdsh -N NodeList rpm -e gpfs.crypto-5.1.5-x.x86_64
mmdsh -N NodeList rpm -e gpfs.adv-5.1.5-x.x86_64
mmdsh -N NodeList rpm -e gpfs.msg.en_US-5.1.5-x.noarch
mmdsh -N NodeList rpm -e gpfs.gskit-8.0.55.x.x86_64
```

```
mmdsh -N NodeList rpm -e gpfs.compression-5.1.5-x.x86_64
mmdsh -N NodeList rpm -e gpfs.license.xx-5.1.5-x.x86_64
mmdsh -N NodeList rpm -e gpfs.docs-5.1.5-x.noarch
mmdsh -N NodeList rpm -e gpfs.gpl-5.1.5-x.noarch
mmdsh -N NodeList rpm -e gpfs.base-5.1.5-x.x86_64
```

Ubuntu

```
mmdsh -N NodeList dpkg -P gpfs.base-debuginfo_5.1.5-x.x86_64
mmdsh -N NodeList dpkg -P gpfs.crypto_5.1.5-x.x86_64
mmdsh -N NodeList dpkg -P gpfs.adv_5.1.5-x.x86_64
mmdsh -N NodeList dpkg -P gpfs.msg.en-us_5.1.5-x.all
mmdsh -N NodeList dpkg -P gpfs.gskit_8.0.55.x.x86_64
mmdsh -N NodeList dpkg -P gpfs.compression_5.1.5-x.x86_64
mmdsh -N NodeList dpkg -P gpfs.license.xx_5.1.5-x.x86_64
mmdsh -N NodeList dpkg -P gpfs.docs_5.1.5-x.noarch
mmdsh -N NodeList dpkg -P gpfs.gpl_5.1.5-x.noarch
mmdsh -N NodeList dpkg -P gpfs.base_5.1.5-x.x86_64
```

Note: In the preceding commands, the package name `gpfs.license.xx` needs to be changed depending on the IBM Spectrum Scale product edition.

13. Reinstall GPFS.

14. Proceed to [“Installation prerequisites”](#) on page 352 and [“Using the installation toolkit to perform installation tasks: Explanations and examples”](#) on page 407.

Note: If you want to remove all cluster configurations, you can also apply the `mmde1node -f` command to each node; however, if you choose to do so, you will also have to recreate cluster, BSDs, and file systems.

Uninstalling the performance monitoring tool

You can uninstall the performance monitoring tool by running the following commands on all nodes that have monitoring enabled on it.

RHEL or SLES nodes:

- To uninstall the sensor on the node, use the `rpm -evh gpfs.gss.pmsensors` command.
- To uninstall the collector on the node, use the `rpm -evh gpfs.gss.pmcollector` command.
- To uninstall the object proxy on the node, use the `rpm -evh pmswift` command.
- To uninstall the NFS proxy on the node, use the `rpm -evh gpfs.pm-ganesha` command.

Ubuntu nodes:

- To uninstall the sensor on the node, use the `dpkg -r gpfs.gss.pmsensors` command.
- To uninstall the collector on the node, use the `dpkg -r gpfs.gss.pmcollector` command.
- To uninstall the object proxy on the node, use the `dpkg -r pmswift` command.
- To uninstall the NFS proxy on the node, use the `dpkg -r gpfs.pm-ganesha` command.

For reinstalling sensors and the collector, see [“Manually installing the performance monitoring tool”](#) on page 376.

Uninstalling the IBM Spectrum Scale management GUI

Do the following to uninstall management GUI and remove the performance monitoring components that are installed for the GUI:

1. Issue the `systemctl stop` command as shown in the following example:

```
systemctl stop gpfsGUI
```

2. If the GUI runs on an IBM Spectrum Scale cluster where sudo wrapper is enabled, export the name of the user that was configured as the file system administrator as an environment variable by issuing the following command:

```
export SUDO_USER=gpfsadmin
```

In this example, the name of the sudo user is *gpfsadmin*. Exporting the environment variable is necessary so that the uninstall process get to know which user name must be used to run the administrative file system commands that are necessary while uninstalling the GUI. For more information on how to configure the IBM Spectrum Scale GUI to use sudo wrapper, see *Configuring IBM Spectrum Scale(tm) GUI to use sudo wrapper* in *IBM Spectrum Scale: Administration Guide*.

3. Issue the following command to clean up the GUI database:

```
psql postgres postgres -c "drop schema fscc cascade"
```

4. Issue one of the following commands depending on the platform to remove the GUI package.

- Red Hat Enterprise Linux and SLES

```
rpm -e gpfs.gui-5.1.5-x.noarch
```

- Ubuntu

```
dpkg -r gpfs.gui_5.1.5-x_all.deb
```

Removing nodes from management GUI-related node class

If you are reinstalling IBM Spectrum Scale, in some scenarios you might need to remove some nodes from the node classes that are used by the management GUI.

For information about these node classes, see [“Node classes used for the management GUI” on page 385](#). If you want to remove the GUI designation of a node, you must remove it from the `GUI_MGT_SERVERS` node class. After you remove a node from this node class, it is no longer designated as a GUI node and GUI services are not started on this node after reinstalling IBM Spectrum Scale. To remove a node from the `GUI_MGT_SERVERS` node class, use the **mmchnodeclass** command only when the GUI services are not running.

On the node that you want to remove from the `GUI_MGT_SERVERS` node class, run the following commands:

```
systemctl stop gpfsGUI
mmchnodeclass GUI_MGMT_SERVERS delete -N guinode
```

These commands stop the GUI services and remove the GUI node from the `GUI_MGT_SERVERS` node class.

Note: If you use **mmchnodeclass** to change the `GUI_MGT_SERVERS` node class while the GUI services are running, the management GUI adds the removed node to the `GUI_MGT_SERVERS` node class again.

Permanently uninstall Cloud services and clean up the environment

Before setting up a fresh multi-node, you must clean up the environment.

Perform the following steps to do a cleanup of the environment and be able to create a fresh multi-node setup.

1. Recall any data that is migrated to the cloud storage by Transparent cloud tiering. For more information, see the *Recalling files from the cloud storage tier* topic in the *IBM Spectrum Scale: Administration Guide*.
2. Delete the container pair set by issuing a command similar to this:

```
mmcloudgateway containerpairset delete --cloud-nodeclass cloud1 --container-pair-set-name vm206
```

For more information, see *Creating a container pair set* topic in the *IBM Spectrum Scale: Administration Guide*.

3. Delete the Cloud services by issuing a command similar to this:

```
mmcloudgateway cloudservice delete --cloud-nodeclass cloud --cloud-service-name newServ
```

For more information, see *Creating a Cloud services* topic in the *IBM Spectrum Scale: Administration Guide*.

4. Delete the CSAP by issuing a command similar to this:

```
mmcloudgateway cloudStorageAccessPoint delete --cloud-nodeclass cloud --cloud-storage-access-point-name csap1
```

For more information, see *Defining cloud storage access points* topic in the *IBM Spectrum Scale: Administration Guide*.

5. Delete the cloud storage account by issuing a command similar to this:

```
mmcloudgateway account delete --cloud-nodeclass cloud --account-name new
```

For more information, see the *Managing a cloud storage account* topic in the *IBM Spectrum Scale: Administration Guide*.

6. Stop the Cloud services on all nodes by issuing a command similar to this:

```
mmcloudgateway service stop -N cloud1
```

For more information, see the *Stopping the Transparent cloud tiering* topic in the *IBM Spectrum Scale: Administration Guide*.

7. Disable Cloud services nodes on the node group by issuing a command similar to this:

```
mmchnode --cloud-gateway-disable -N vm1,vm2 --cloud-gateway-nodeclass cloud1
```

For more information, see the *Designating the Transparent cloud tiering nodes* topic in the *IBM Spectrum Scale: Administration Guide*.

8. Delete the node class if required. For more information, see the **mmdelnodeclass** command in the *IBM Spectrum Scale: Command and Programming Reference*
9. Uninstall the Cloud services RPMs (both server and client) from all the nodes by issuing the following command:

- `rpm -e gpfs.tct.server-x.x.x.x86_64`
- `rpm -e gpfs.tct.client-x.x.x.x86_64.rpm`

Chapter 16. Upgrading

To upgrade to a newer version of IBM Spectrum Scale, first consider the version that you are upgrading from and then consider coexistence and compatibility issues.

IBM Spectrum Scale supports a limited form of backward compatibility between two adjacent releases. Limited backward compatibility allows you to temporarily operate with a mixture of IBM Spectrum Scale nodes running on the newer version and nodes running an earlier version:

- Within a cluster, limited backward compatibility enables you to perform an online upgrade to the new IBM Spectrum Scale version of the code if an online upgrade from your current version to the newer version is supported.
- In a multicluster environment, limited backward compatibility allows the individual clusters to be upgraded on their own schedules. Access to the file system data can be preserved even though some of the clusters might still be running at an earlier version.
- In multicluster environments, it is recommended to upgrade the home cluster before the cache cluster especially if file audit logging, watch folder, clustered watch, and AFM functions are being used.

Tip: Ensure that there is approximately 5 GB of free space on the nodes to download, extract, and install the packages.

Note: Cluster Export Services (CES) nodes that have the Samba service enabled, do not support backward compatibility. For more information about upgrading CES nodes, see [“Upgrade process flow”](#) on page 545.

Online upgrades allow you to install new IBM Spectrum Scale code on one node at a time while maintaining access to the file systems without shutting down IBM Spectrum Scale on other nodes. However, you must upgrade all nodes within a short time. The time dependency exists because some IBM Spectrum Scale features in the newer version become available on each node as soon as the node is upgraded, while other features are not available until you upgrade all participating nodes.

You can also perform offline upgrades, if you can shut down the entire cluster. An offline upgrade is similar to the online upgrade procedure. However, due to the entire cluster being offline, it is possible to jump straight to the latest code level instead of hopping to an intermediate level, as might be required during an online upgrade. For information about an offline upgrade procedure, see [“Offline upgrade with complete cluster shutdown”](#) on page 590.

Important: Online upgrade from an unsupported version of IBM Spectrum Scale or GPFS is not supported.

For information about the upgrade process flow used by the installation toolkit, see [“Upgrade process flow”](#) on page 545. You can use this process flow as a reference for doing manual upgrades on Linux nodes.

After all nodes are upgraded to the new code, you must finalize the upgrade by running the commands **mmchconfig release=LATEST** and **mmchfs -V full** (or **mmchfs -V compat**). Also, certain new features might require you to run the **mmmigratefs** command to enable them. For more information, see [“Completing the upgrade to a new level of IBM Spectrum Scale”](#) on page 572.

For the latest information on upgrade, coexistence, and compatibility, see the *Supported OS and software versions* section of [IBM Spectrum Scale FAQ](#) in IBM Documentation.

IBM Spectrum Scale upgrade consists of these topics:

- [“IBM Spectrum Scale supported upgrade paths”](#) on page 506
- [“Online upgrade support for protocols and performance monitoring”](#) on page 507
- [“Upgrading IBM Spectrum Scale nodes”](#) on page 508
- [“Upgrading IBM Spectrum Scale non-protocol Linux nodes”](#) on page 509
- [“Upgrading IBM Spectrum Scale protocol nodes”](#) on page 512

- [“Upgrading AFM and AFM DR” on page 518](#)
- [“Upgrading call home” on page 524](#)
- [“Upgrading object packages” on page 520](#)
- [“Upgrading SMB packages” on page 521](#)
- [“Upgrading NFS packages” on page 523](#)
- [“Manually upgrading pmswift” on page 527](#)
- [“Manually upgrading the performance monitoring tool” on page 526](#)
- [“Manually upgrading the IBM Spectrum Scale management GUI” on page 529](#)
- [“Upgrading Cloud services” on page 531](#)
- [“Upgrading to IBM Cloud Object Storage software level 3.7.2 and above” on page 541](#)
- [“Upgrading IBM Spectrum Scale components with the installation toolkit” on page 543](#)
- [“Changing the IBM Spectrum Scale product edition” on page 566](#)
- [“Completing the upgrade to a new level of IBM Spectrum Scale” on page 572](#)
- [“Reverting to the previous level of IBM Spectrum Scale” on page 577](#)
- [“Coexistence considerations” on page 579](#)
- [“Compatibility considerations” on page 579](#)
- [“Considerations for IBM Spectrum Protect for Space Management” on page 579](#)
- [“Applying maintenance to your IBM Spectrum Scale system” on page 580](#)
- [“Guidance for upgrading the operating system on IBM Spectrum Scale nodes” on page 580](#)
- [“Considerations for upgrading from an operating system not supported in IBM Spectrum Scale 5.1.x.x” on page 586](#)
- [“Servicing IBM Spectrum Scale protocol nodes” on page 589](#)
- [“Offline upgrade with complete cluster shutdown” on page 590](#)

IBM Spectrum Scale supported upgrade paths

Use this information to understand the supported upgrade paths for IBM Spectrum Scale.

The following table lists the supported online upgrade paths for IBM Spectrum Scale.

Table 44. IBM Spectrum Scale supported online upgrade paths												
Upgrading from:	To: 5.0.0.x	To: 5.0.1.x	To: 5.0.2.x	To: 5.0.3.x	To: 5.0.4.x	To: 5.0.5.x	To: 5.1.0.x	To: 5.1.1.x	To: 5.1.2.x	To: 5.1.3.x	To: 5.1.4.x	To: 5.1.5.x
5.0.0.x	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
5.0.1.x	--	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
5.0.2.x	--	--	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
5.0.3.x	--	--	--	✓	✓	✓	✓	✓	✓	✓	✓	✓
5.0.4.x	--	--	--	--	✓	✓	✓	✓	✓	✓	✓	✓
5.0.5.x	--	--	--	--	--	✓	✓	✓	✓	✓	✓	✓
5.1.0.x	--	--	--	--	--	--	✓	✓	✓	✓	✓	✓
5.1.1.x	--	--	--	--	--	--	--	✓	✓	✓	✓	✓
5.1.2.x	--	--	--	--	--	--	--	--	✓	✓	✓	✓
5.1.3.x	--	--	--	--	--	--	--	--	--	✓	✓	✓

Table 44. IBM Spectrum Scale supported online upgrade paths (continued)												
Upgrading from:	To: 5.0.0. x	To: 5.0.1. x	To: 5.0.2. x	To: 5.0.3. x	To: 5.0.4. x	To: 5.0.5. x	To: 5.1.0. x	To: 5.1.1. x	To: 5.1.2. x	To: 5.1.3. x	To: 5.1.4. x	To: 5.1.5. x
5.1.4.x	--	--	--	--	--	--	--	--	--	--	✓	✓
5.1.5.x	--	--	--	--	--	--	--	--	--	--	--	✓

- ✓: Supported
- X: Not supported
- --: Not applicable

Note: After an IBM Spectrum Scale version is past its end of service (EOS) date, no further testing or proof of upgrade path to later supported releases is done. It is strongly recommended to upgrade to a newer version of IBM Spectrum Scale that is supported before your currently installed version is past its EOS date.

Important: The versions that are not shown in the upgrade paths table are past their EOS date. You can access upgrade procedures for these versions in the IBM Spectrum Scale documentation of earlier releases. For example, if you have IBM Spectrum Scale 4.2.3.x version, refer to IBM Spectrum Scale 5.0.x documentation. However, online upgrade from an unsupported version is not supported. You must use an offline upgrade for such upgrade paths.

Remember:

- Online upgrade support for protocol nodes depends on which services have been enabled and deployed. Different protocol services have different levels of online upgrade support. For more information, see [“Online upgrade support for protocols and performance monitoring”](#) on page 507.
- IBM Spectrum Scale installation toolkit can be used for upgrades from version 4.1.1.x onwards.

Related information

[IBM Spectrum Scale Software Version Recommendation Preventive Service Planning](#)
[ESS supported upgrade paths](#)

Online upgrade support for protocols and performance monitoring

Use this information to understand the online upgrade support for protocols and performance monitoring in IBM Spectrum Scale.

Online upgrade support for protocol nodes depends on which services have been enabled and deployed. Different protocol services have different levels of online upgrade support.

- NFS: If there are different NFS versions on protocol nodes in a cluster, then performing an online upgrade on one node at a time can be tolerated. .
- SMB: All protocol nodes running the SMB service must have the same version installed at any time. In addition to a brief outage of the SMB service, this also requires a brief outage of the NFS service in case Active Directory (AD) authentication is being used. SMB upgrade in an IBM Spectrum Scale cluster is done in two phases. The protocol nodes in the cluster are divided in two halves and SMB is upgraded on each half one by one.
- Object: All object nodes must be at the same level as object operations interact with all nodes. Therefore, object services need to be stopped across the cluster for the duration of the upgrade. Other protocol services can continue to be run.
- Performance monitoring: Having two different versions on collector nodes will result in a loss of communication and could potentially result in missing performance metrics until collector versions match. This limitation is not applicable if the source and the target versions are later than 5.0.1.

Upgrading IBM Spectrum Scale nodes

IBM Spectrum Scale supports upgrade to the latest release from a previous supported IBM Spectrum Scale release. For more information, see [“IBM Spectrum Scale supported upgrade paths”](#) on page 506 and [“Online upgrade support for protocols and performance monitoring”](#) on page 507.

To upgrade an IBM Spectrum Scale cluster, do the following steps for each node in the cluster.

For upgrading Linux nodes, use the following procedures.

- [“Upgrading IBM Spectrum Scale non-protocol Linux nodes”](#) on page 509.
- [“Upgrading IBM Spectrum Scale protocol nodes”](#) on page 512.

For upgrading Hadoop, use the following procedures.

CDP Private Cloud Base distribution

- To upgrade the CDP Private Cloud Base distribution, see *Upgrading CDP* in *IBM Spectrum Scale: Big Data and Analytics Guide*.

HDP Ambari distribution

- To upgrade the IBM Spectrum Scale Hadoop Connector through Ambari, see *Upgrading HDFS Transparency* in *IBM Spectrum Scale: Big Data and Analytics Guide*.
- To upgrade the IBM Spectrum Scale Hadoop Connector manually, see *Upgrading the HDFS Transparency cluster* in *IBM Spectrum Scale: Big Data and Analytics Guide*.

For upgrading AIX and Windows nodes, use the following steps.

1. Stop all user activity in the file systems on the node that you are upgrading.
2. If the call home service is enabled on the cluster, issue the following command to disable it:

```
mmcallhome capability disable
```

3. **Important:** Back up your file systems to ensure that your data is protected.
4. If file systems are mounted on the node, issue the **mmumount** command to unmount them.
Do not use the **-f** option of the **mmumount** command.
5. Shut down IBM Spectrum Scale on the node by issuing the following command:

```
mmshutdown
```

Note: You do not need to close **mmccrmonitor** and other mm processes because they are handled by IBM Spectrum Scale upgrade post-install scripts.

6. Upgrade the node to new version of IBM Spectrum Scale. The steps for upgrading depend on the operating system that is installed on the node.

• Instructions for upgrading AIX nodes:

- When you upgrade from 5.0.x or later to 5.1.y.z, you must first upgrade to 5.1.y.0 and then upgrade to 5.1.y.z.
- Copy the installation images and install the IBM Spectrum Scale licensed programs as described in the [Chapter 5, “Installing IBM Spectrum Scale on AIX nodes,”](#) on page 453 section.

• Instructions for upgrading Windows nodes:

- a. Open the **Programs and Features** window and remove **IBM Spectrum Scale Standard Edition 5.0.y**.
 - i) Uninstall the version of IBM Spectrum Scale from which you are upgrading and reboot the system.
 - ii) Uninstall IBM GPFS GSKit applicable for the IBM Spectrum Scale version from which you are upgrading and reboot the system.

- iii) Uninstall the IBM Spectrum Scale license.
 - b. Copy the installation images and install the IBM Spectrum Scale licensed program as described in [Installing IBM Spectrum Scale on Windows nodes](#).
- Important:** For Windows upgrades, you must issue all the IBM Spectrum Scale administration commands from the 5.1.x node.
7. Start IBM Spectrum Scale on the node by issuing the following command:

```
mmstartup
```

8. Mount any file systems that are not mounted automatically when the IBM Spectrum Scale daemon starts.

When all the nodes in the cluster are successfully upgraded to the new level, the next step is to complete the upgrade. For more information, see [“Completing the upgrade to a new level of IBM Spectrum Scale”](#) on page 572.

Upgrading IBM Spectrum Scale non-protocol Linux nodes

Use these steps to upgrade non-protocol nodes that are running on Linux distributions in an IBM Spectrum Scale cluster.

Do the following steps for each non-protocol node in the cluster.

1. Stop all user activity in the file systems on the node that you are upgrading.
2. If the call home service is enabled on the cluster, run the following command to disable it:

```
mmcallhome capability disable
```

Important: Back up your file systems to ensure that your data is protected.

3. If file systems are mounted on the node, run the following command to unmount them:

```
mmumount all
```

Do not use the **-f** option of the **mmumount** command.

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.

4. Shut down IBM Spectrum Scale on the node by running the following command.

```
mmshutdown
```

Note: You do not need to close the **mmccrmonitor** command and other mm processes because they are handled by IBM Spectrum Scale upgrade post-install scripts.

Verify that the GPFS daemon has terminated and that the kernel extensions are unloaded by using **mmfsenv -u**. If the **mmfsenv -u** command reports that it cannot unload the kernel extensions because they are busy, then the installation can proceed, but the node must be rebooted after the installation. Kernel extensions are busy means that a process has a current directory in some GPFS file system directory or it has an open file descriptor. The Linux utility **lssof** can identify the process and that process can then be killed. Retry **mmfsenv -u** after killing the process and if the command succeeds, then a reboot of the node can be avoided.

Note:

- The **mmfsenv -u** command is normally used with assistance from support or for cases where its usage is specifically documented such as in this scenario.
 - The **mmfsenv -u** command only checks the local node from which it is run, not all the nodes in the cluster.
5. Extract the IBM Spectrum Scale software packages as described in the topic [“Extracting the IBM Spectrum Scale software on Linux nodes”](#) on page 355. For information about the location of extracted packages, see [“Location of extracted packages”](#) on page 361.

6. Change to the directory that contains the IBM Spectrum Scale package files.

- If Red Hat Enterprise Linux or SUSE Linux Enterprise Server (SLES) is installed on the node, run the following command:

```
cd /usr/lpp/mmfs/<x.x.x.x>/gpfs_rpms
```

- If Ubuntu Linux is installed on the node, run the following command:

```
cd /usr/lpp/mmfs/<x.x.x.x>/gpfs_debs
```

Where `<x.x.x.x>` is the version level of IBM Spectrum Scale that you are upgrading to.

7. Upgrade the node by using one of the following set of commands. The command choice depends on the operating system and the product edition that is installed on the node.

Note: The following commands are specific to the current release. If you are upgrading to an earlier release, see the documentation for that release.

Tip: To determine the currently installed edition, run the following command:

```
mmlslicense -L
```

Note:

- Starting with IBM Spectrum Scale 5.1.0 or later, there are no separate packages for call home, since call home is integrated into the `gpfs.base` package.
- If you are upgrading to IBM Spectrum Scale 5.1.0 or later, you do not need to upgrade the `gpfs.kafka` package.
- Red Hat Enterprise Linux 7.x and 8.x
 - If IBM Spectrum Scale Standard Edition or IBM Spectrum Scale Data Access Edition is installed on the node, run the following command. The entire command must be on one line.

```
rpm -Fvh ./gpfs.gskit*.rpm ./gpfs.gpl*.rpm ./gpfs.docs*.rpm ./gpfs.compression*.rpm  
./gpfs.base*.rpm ./gpfs.msg.en_US*.rpm  
./gpfs.java*.rpm ./rhel/gpfs.librdkafka*.rpm
```

- If IBM Spectrum Scale Advanced Edition or IBM Spectrum Scale Data Management Edition or IBM Spectrum Scale Developer Edition is installed on the node, run the following command. The entire command must be on one line:

```
rpm -Fvh ./gpfs.msg.en_US*.rpm ./gpfs.java*.rpm ./gpfs.gskit*.rpm  
./gpfs.gpl*.rpm ./gpfs.docs*.rpm ./gpfs.crypto*.rpm ./gpfs.compression*.rpm  
./gpfs.base*.rpm ./gpfs.adv*.rpm ./rhel/gpfs.librdkafka*.rpm
```

- SUSE Linux Enterprise Server (SLES)
 - If IBM Spectrum Scale Standard Edition or IBM Spectrum Scale Data Access Edition is installed on the node, run the following command. The entire command must be on one line:

```
rpm -Fvh ./gpfs.gskit*.rpm ./gpfs.gpl*.rpm ./gpfs.docs*.rpm ./gpfs.compression*.rpm  
./gpfs.base*.rpm ./gpfs.msg.en_US*.rpm  
./gpfs.java*.rpm
```

- If IBM Spectrum Scale Advanced Edition or IBM Spectrum Scale Data Management Edition is installed on the node, run the following command. The entire command must be on one line:

```
rpm -Fvh ./gpfs.msg.en_US*.rpm ./gpfs.java*.rpm ./gpfs.gskit*.rpm  
./gpfs.gpl*.rpm ./gpfs.docs*.rpm ./gpfs.crypto*.rpm ./gpfs.compression*.rpm  
./gpfs.base*.rpm ./gpfs.adv*.rpm
```

- Ubuntu Linux

- If IBM Spectrum Scale Standard Edition or IBM Spectrum Scale Data Access Edition is installed on the node, run the following command. The entire command must be on one line:

```
apt-get install --only-upgrade ./gpfs.gskit*.deb ./gpfs.gpl*.deb
./gpfs.docs.deb ./gpfs.compression.deb
./gpfs.base*.deb ./gpfs.msg.en-us*.deb
./gpfs.java*.deb ./ubuntu/gpfs.librdkafka*.deb
```

- If IBM Spectrum Scale Advanced Edition or IBM Spectrum Scale Data Management Edition is installed on the node, run the following command. The entire command must be on one line:

```
apt-get install --only-upgrade ./gpfs.gskit*.deb ./gpfs.gpl*.deb
./gpfs.docs*.deb ./gpfs.crypto*.deb ./gpfs.compression*.deb
./gpfs.base*.deb ./gpfs.adv*.deb
./gpfs.msg.en-us*.deb ./gpfs.java*.deb
./ubuntu/gpfs.librdkafka*.deb
```

8. Install the following packages if needed.

- a. For all editions, if you are upgrading from IBM Spectrum Scale 4.2.3.x or earlier, install the `gpfs.compression` package:

- If Red Hat Enterprise Linux or SUSE Linux Enterprise Server Linux is installed on the node, run the following command from the `gpfs_rpms` directory:

```
rpm -ivh gpfs.compression*.rpm
```

- If Ubuntu Linux is installed on the node, run the following command from the `gpfs_debs` directory:

```
apt-get install gpfs.compression*.deb
```

- b. For all editions, for the `gpfs.ext` package:

- If you are upgrading to IBM Spectrum Scale 5.0.1 or earlier, ensure that you upgrade the `gpfs.ext` package.
- In versions 5.0.2 and later, you do not need to upgrade the `gpfs.ext` package. This package is removed and its functions are merged into other packages.



Attention: On Ubuntu Linux nodes, in version 5.0.2 and later, you might see an older version of the `gpfs.ext` package. You can safely remove this package by running the following command:

```
dpkg -P gpfs.ext
```

- c. For IBM Spectrum Scale Advanced Edition and IBM Spectrum Scale Data Management Edition, if you plan to use the AFM-based Asynchronous Disaster Recovery (AFM DR) feature, the file encryption feature, or the Transparent cloud tiering feature, you must install the following packages:

- `gpfs.adv`
- `gpfs.crypto`

9. After a successful upgrade, rebuild the GPFS portability layer (GPL). For more information, see [“Building the GPFS portability layer on Linux nodes” on page 364](#).

10. Upgrade performance monitoring packages. For more information, see [“Manually upgrading the performance monitoring tool” on page 526](#) and [“Manually upgrading pmswift” on page 527](#).

Note: You might see the error message `pmcollector service critical error` on GUI nodes. If this problem occurs, restart the `pmcollector` service by running the following command on all GUI nodes:

```
systemctl restart pmcollector
```

11. Upgrade management GUI packages. For more information, see [“Manually upgrading the IBM Spectrum Scale management GUI”](#) on page 529.
12. Start IBM Spectrum Scale on the node by running the following command:

```
mmstartup
```

13. Mount any file systems that are not mounted automatically when the IBM Spectrum Scale daemon starts.
14. If you disable the call home service in a preceding step, run the following command to enable it:

```
mmcallhome capability enable accept
```

Repeat the preceding steps on all non-protocol nodes in the cluster. After these steps are done for all non-protocol nodes in the cluster, proceed with the following steps.

15. Upgrade the license package corresponding to the installed IBM Spectrum Scale product edition on all nodes in the cluster.

- For Red Hat Enterprise Linux and SLES:

```
mmdsh -N NodeList rpm -Fvh /usr/lpp/mmfs/N.N.N.N/gpfs_rpms/gpfs.license-xxx*.rpm
```

- For Ubuntu:

```
mmdsh -N NodeList apt-get install --only-upgrade /usr/lpp/mmfs/N.N.N.N/gpfs_debs/  
gpfs.license-xxx*.deb
```

- *NodeList* is the list of all nodes in the cluster.
- *N.N.N.N* is the product version. For example, 5.0.2.0.
- *xxx* is the indicator of the product edition:
 - da: Data Access Edition
 - dm: Data Management Edition
 - dev: Developer Edition
 - std: Standard Edition
 - adv: Advanced Edition

16. If the product edition is Advanced, Data Management, or Developer and file audit logging or clustered watch folder is enabled in the cluster, see [“Upgrade paths and commands for file audit logging and clustered watch folder”](#) on page 542.

- If you are upgrading to IBM Spectrum Scale 5.1.2 or later and you have previously run the **mmmsgqueue config --remove-msgqueue** command, no further action is required.
- If you are upgrading to IBM Spectrum Scale 5.1.2 or later and you have previously not run the **mmmsgqueue config --remove-msgqueue** command, do one of the following steps.
 - If you have never enabled the message queue, manually remove the `gpfs.kafka` package.
 - In all other cases, run the **mmmsgqueue config --remove-msgqueue** command. This command removes the `gpfs.kafka` package at the end of the cleanup.

When all the nodes in the cluster are successfully upgraded to the new level, the next step is to complete the upgrade. For more information, see [“Completing the upgrade to a new level of IBM Spectrum Scale”](#) on page 572.

Upgrading IBM Spectrum Scale protocol nodes

Use these steps to upgrade protocol nodes in an IBM Spectrum Scale cluster.

Perform the following steps for each protocol node in the cluster.

1. Stop all user activity in the file systems on the node that you are upgrading.
2. Stop any protocol services that are running on the node:

- a. Issue the following command to suspend Cluster Export Services (CES) and stop the protocol services on the node:

```
mmces node suspend --stop
```

If you are upgrading from IBM Spectrum Scale version 5.0.2.0 or earlier, issue the following commands to suspend the protocol node and stop the protocol services:

```
mmces node suspend  
mmces service stop Protocol
```

- b. If you are upgrading from IBM Spectrum Scale version 5.0.1.2 or earlier to version 5.0.2 or later, issue the following command to ensure that the **mmumount** of the CES file system completes successfully:

```
mmcesmonitor stop
```

3. If the call home service is enabled on the cluster, issue the following command to disable it:

```
mmcallhome capability disable
```

Important: Back up your file systems to ensure that your data is protected.

4. If file systems are mounted on the node, issue the following command to unmount them:

```
mmumount all
```

Do not use the **-f** option of the **mmumount** command.

5. Shut down IBM Spectrum Scale on the node by issuing the following command:

```
mmshutdown
```

Note: You do not need to close **mmccrmonitor** and other mm processes because they are handled by IBM Spectrum Scale upgrade post-install scripts.

Verify that the GPFS daemon has terminated and that the kernel extensions are unloaded by using **mmfsenv -u**. If the **mmfsenv -u** command reports that it cannot unload the kernel extensions because they are busy, then the installation can proceed, but the node must be rebooted after the installation. Kernel extensions are busy means that a process has a current directory in some GPFS file system directory or it has an open file descriptor. The Linux utility **lssof** can identify the process and that process can then be killed. Retry **mmfsenv -u** after killing the process and if the command succeeds, then a reboot of the node can be avoided.

Note:

- The **mmfsenv -u** command is normally used with assistance from support or for cases where its usage is specifically documented such as in this scenario.
 - The **mmfsenv -u** command only checks the local node from which it is run, not all the nodes in the cluster.
6. Extract the IBM Spectrum Scale software packages as described in the topic [“Extracting the IBM Spectrum Scale software on Linux nodes”](#) on page 355. For information about the location of extracted packages, see [“Location of extracted packages”](#) on page 361.
 7. Change to the directory that contains the IBM Spectrum Scale package files.
 - If Red Hat Enterprise Linux or SUSE Linux Enterprise Server (SLES) is installed on the node, issue the following command:

```
cd /usr/lpp/mmfs/<x.x.x.x>/gpfs_ipms
```

- If Ubuntu Linux is installed on the node, issue the following command:

```
cd /usr/lpp/mmfs/<x.x.x.x>/gpfs_debs
```

Where **<x.x.x.x>** is the version level of IBM Spectrum Scale that you are upgrading to.

8. Upgrade the protocol node by using one of the following set of steps for depending on the operating system and the product edition that is installed on the node.

Note: The following commands are specific to the current release. If you are upgrading to an earlier release, see the documentation for that release.

Tip: To determine the currently installed edition, issue the following command:

```
mmlslicense -L
```

Note:

- Starting with IBM Spectrum Scale 5.1.0 or later, there are no separate packages for call home, since call home is integrated into the `gpfs.base` package.
- If you are upgrading to IBM Spectrum Scale 5.1.0 or later, you do not need to upgrade the `gpfs.kafka` package.
- Red Hat Enterprise Linux 7.x and 8.x
 - If IBM Spectrum Scale Standard Edition or IBM Spectrum Scale Data Access Edition is installed on the node, run the following command. The entire command must be on one line.

```
rpm -Fvh ./gpfs.gskit*.rpm ./gpfs.gpl*.rpm ./gpfs.docs*.rpm ./gpfs.compression*.rpm  
./gpfs.base*.rpm ./gpfs.msg.en_US*.rpm  
./gpfs.java*.rpm ./rhel/gpfs.librdkafka*.rpm
```

- If IBM Spectrum Scale Advanced Edition or IBM Spectrum Scale Data Management Edition or IBM Spectrum Scale Developer Edition is installed on the node, run the following command. The entire command must be on one line:

```
rpm -Fvh ./gpfs.msg.en_US*.rpm ./gpfs.java*.rpm ./gpfs.gskit*.rpm  
./gpfs.gpl*.rpm ./gpfs.docs*.rpm ./gpfs.crypto*.rpm ./gpfs.compression*.rpm  
./gpfs.base*.rpm ./gpfs.adv*.rpm ./rhel/gpfs.librdkafka*.rpm
```

- SUSE Linux Enterprise Server (SLES)
 - If IBM Spectrum Scale Standard Edition or IBM Spectrum Scale Data Access Edition is installed on the node, run the following command. The entire command must be on one line:

```
rpm -Fvh ./gpfs.gskit*.rpm ./gpfs.gpl*.rpm ./gpfs.docs*.rpm ./gpfs.compression*.rpm  
./gpfs.base*.rpm ./gpfs.msg.en_US*.rpm  
./gpfs.java*.rpm
```

- If IBM Spectrum Scale Advanced Edition or IBM Spectrum Scale Data Management Edition is installed on the node, run the following command. The entire command must be on one line:

```
rpm -Fvh ./gpfs.msg.en_US*.rpm ./gpfs.java*.rpm ./gpfs.gskit*.rpm  
./gpfs.gpl*.rpm ./gpfs.docs*.rpm ./gpfs.crypto*.rpm ./gpfs.compression*.rpm  
./gpfs.base*.rpm ./gpfs.adv*.rpm
```

- Ubuntu Linux
 - If IBM Spectrum Scale Standard Edition or IBM Spectrum Scale Data Access Edition is installed on the node, run the following command. The entire command must be on one line:

```
apt-get install --only-upgrade ./gpfs.gskit*.deb ./gpfs.gpl*.deb  
./gpfs.docs.deb ./gpfs.compression.deb  
./gpfs.base*.deb ./gpfs.msg.en-us*.deb  
./gpfs.java*.deb ./ubuntu/gpfs.librdkafka*.deb
```

- If IBM Spectrum Scale Advanced Edition or IBM Spectrum Scale Data Management Edition is installed on the node, run the following command. The entire command must be on one line:

```
apt-get install --only-upgrade ./gpfs.gskit*.deb ./gpfs.gpl*.deb  
./gpfs.docs*.deb ./gpfs.crypto*.deb ./gpfs.compression*.deb  
./gpfs.base*.deb ./gpfs.adv*.deb  
./gpfs.msg.en-us*.deb ./gpfs.java*.deb  
./ubuntu/gpfs.librdkafka*.deb
```

9. If needed, install or upgrade the following packages.

- a. For all editions, if you are upgrading from IBM Spectrum Scale 4.2.3.x or earlier, install the `gpfs.compression` package:

- If Red Hat Enterprise Linux or SUSE Linux Enterprise Server Linux is installed on the node, issue the following command from the `gpfs_rpms` directory:

```
rpm -ivh gpfs.compression*.rpm
```

- If Ubuntu Linux is installed on the node, issue the following command from the `gpfs_debs` directory:

```
apt-get install gpfs.compression*.deb
```

b. For all editions, for the `gpfs.ext` package:

- If you are upgrading to IBM Spectrum Scale 5.0.1 or earlier, ensure that you upgrade the `gpfs.ext` package.
- In versions 5.0.2 and later, you do not need to upgrade the `gpfs.ext` package. The package is not present because its functionality is merged into other packages.



Attention: On Ubuntu Linux nodes, in version 5.0.2 and later, you might see an older version of the `gpfs.ext` package. You can safely remove this package by issuing the following command:

```
dpkg -P gpfs.ext
```

- c. For IBM Spectrum Scale Advanced Edition and IBM Spectrum Scale Data Management Edition, if you plan to use the AFM-based Asynchronous Disaster Recovery feature, the file encryption feature, or the Transparent cloud tiering feature, you must install the following packages:

- `gpfs.adv`
- `gpfs.crypto`

10. After a successful upgrade, rebuild the GPFS portability layer (GPL). For more information, see [“Building the GPFS portability layer on Linux nodes”](#) on page 364.

11. Upgrade object packages on the protocol node. For more information, see [“Upgrading object packages”](#) on page 520.

12. Upgrade performance monitoring packages. For more information, see [“Manually upgrading the performance monitoring tool”](#) on page 526 and [“Manually upgrading pmswift”](#) on page 527.

Note: You might see the error message `pmcollector service critical error` on GUI nodes. If this problem occurs, restart the `pmcollector` service by issuing the following command on all GUI nodes:

```
systemctl restart pmcollector
```

13. Upgrade management GUI packages. For more information, see [“Manually upgrading the IBM Spectrum Scale management GUI”](#) on page 529.

14. Start IBM Spectrum Scale on the node by issuing the following command:

```
mmstartup
```

15. Mount any file systems that are not mounted automatically when the IBM Spectrum Scale daemon starts.
16. Upgrade NFS packages on the protocol node. For more information, see [“Upgrading NFS packages” on page 523](#).
17. If you disabled the call home service in a preceding step, issue the following command to enable it:

```
mmcallhome capability enable accept
```

18. If you suspended CES in a preceding step, issue the following command to resume CES and start protocol services on the node.

```
mmces node resume --start
```

If you are upgrading from IBM Spectrum Scale version 5.0.2.0 or earlier, issue the following commands to resume the protocol node and start the protocol services:

```
mmces node resume  
mmces service start Protocol
```

Repeat the preceding steps on all protocol nodes in the cluster. After these steps are done for all protocol nodes in the cluster, proceed with the following steps.

19. Do object version sync and start object on all protocol nodes. For more information, see [“Upgrading object packages” on page 520](#).
20. Upgrade SMB packages on all protocol nodes. For more information, see [“Upgrading SMB packages” on page 521](#).
21. If you are upgrading from IBM Spectrum Scale 5.1.0.x or earlier to IBM Spectrum Scale 5.1.1.x or later, remove the `gpfs.protocols-support` package, if it is present.

- For Red Hat Enterprise Linux and SLES:

```
mmdsh -N ProtocolNodeList rpm -ev --nodeps gpfs.protocols-support
```

- For Ubuntu (The entire command must be on one line):

```
mmdsh -N ProtocolNodeList apt-get remove gpfs.protocols-support
```

ProtocolNodeList is the list of all protocol nodes in the cluster.

22. Do the protocol authentication configuration depending on your setup. For more information, see [“Protocol authentication configuration changes during upgrade” on page 563](#).
23. Upgrade the license package corresponding to the installed IBM Spectrum Scale product edition on all nodes in the cluster.

- For Red Hat Enterprise Linux and SLES:

```
mmdsh -N NodeList rpm -Fvh /usr/lpp/mmfs/N.N.N.N/gpfs_rpms/gpfs.license-xxx*.rpm
```

- For Ubuntu:

```
mmdsh -N NodeList apt-get install --only-upgrade /usr/lpp/mmfs/N.N.N.N/gpfs_debs/  
gpfs.license-xxx*.deb
```

- *NodeList* is the list of all nodes in the cluster.
- *N.N.N.N* is the product version. For example, 5.0.2.0.
- *xxx* is the indicator of the product edition:
 - da: Data Access Edition
 - dm: Data Management Edition
 - dev: Developer Edition
 - std: Standard Edition

- adv: Advanced Edition

24. If the product edition is Advanced, Data Management, or Developer and file audit logging or clustered watch folder is enabled in the cluster, see [“Upgrade paths and commands for file audit logging and clustered watch folder”](#) on page 542.

- If you are upgrading to IBM Spectrum Scale 5.1.2 or later and you have previously run the **mmmsgqueue config --remove-msgqueue** command, no further action is required.
- If you are upgrading to IBM Spectrum Scale 5.1.2 or later and you have previously not run the **mmmsgqueue config --remove-msgqueue** command, do one of the following steps.
 - If you have never enabled the message queue, manually remove the `gpfs.kafka` package.
 - In all other cases, run the **mmmsgqueue config --remove-msgqueue** command. This command removes the `gpfs.kafka` package at the end of the cleanup.

When all the nodes in the cluster are successfully upgraded to the new level, the next step is to complete the upgrade. For more information, see [“Completing the upgrade to a new level of IBM Spectrum Scale”](#) on page 572.

Upgrading GPUDirect Storage

You need to upgrade your IBM Spectrum Scale cluster to 5.1.2 or later to start using the GPUDirect Storage (GDS).

There are two types of upgrade paths. The first type is upgrading the clusters that are configured with 5.1.1 technical preview. The next type is upgrading the cluster that is not yet configured with GDS.

Upgrading from 5.1.1 technical preview

The GDS technical preview for IBM Spectrum Scale 5.1.1 and the IBM Spectrum Scale 5.1.2 version are not concurrently upgradeable. You need to perform the following steps when you need to use GDS on IBM Spectrum Scale 5.1.2:

1. Stop all GDS I/O on IBM Spectrum Scale 5.1.1.x.
2. Upgrade IBM Spectrum Scale to 5.1.2 or later. For more information, see [Chapter 16, “Upgrading,”](#) on page 505.
3. Update MOFED to supported level. For more information, see [Mellanox OFED installation](#).
4. Update CUDA to the supported version. For more information, see [Installing GDS](#).
5. Update the NVIDIA GDS driver. For more information, see [GDS package installation](#).

Upgrading from clusters that are not configured with GDS

If you have not used the technical preview in 5.1.1, perform the following steps:

1. Upgrade your cluster to the 5.1.2 (or later) level by upgrading GPFS. For more information, see [Chapter 16, “Upgrading,”](#) on page 505.
2. Proceed through the installation steps that are described in the following section: [Chapter 9, “Installing GPUDirect Storage for IBM Spectrum Scale,”](#) on page 481.

Related concepts

[“GPUDirect Storage support for IBM Spectrum Scale”](#) on page 26

IBM Spectrum Scale's support for NVIDIA's GPUDirect Storage (GDS) enables a direct path between GPU memory and storage. This solution addresses the need for higher throughput and lower latencies. File system storage is directly connected to the GPU buffers to reduce latency and load on CPU. For IBM Spectrum Scale, this means that data can be read directly from an NSD server's pagepool and it is sent to the GPU buffer of the IBM Spectrum Scale clients by using RDMA. IBM Spectrum Scale with GDS requires an InfiniBand or RoCE fabric. In IBM Spectrum Scale, the **mmdiag** command is enhanced to print diagnostic information for GPUDirect Storage.

[“Planning for GPUDirect Storage”](#) on page 250

IBM Spectrum Scale support for GPUDirect Storage (GDS) enables a direct path between GPU memory and storage. You need to ensure that certain conditions are met before you start installing the feature.

Related tasks

[“Installing GPUDirect Storage for IBM Spectrum Scale” on page 481](#)

IBM Spectrum Scale support for GPUDirect Storage (GDS) enables a direct path between GPU memory and storage.

Upgrading AFM and AFM DR

Consider the following while you upgrade Active File Management (AFM) or Active File Management - DR (AFM DR).

Before you upgrade to a newer version of IBM Spectrum Scale, consider the version from which you are upgrading. IBM Spectrum Scale supports a limited form of compatibility with an earlier version between two adjacent releases and hence, coexistence and compatibility measures are required. For more information, see [“IBM Spectrum Scale supported upgrade paths” on page 506](#). The limited compatibility with an earlier version allows temporarily operating with some IBM Spectrum Scale nodes that are running on the newer version, and some nodes that are running an earlier version. Within a cluster, because of this you can perform an online upgrade to the new IBM Spectrum Scale version, if upgrade from your current version to the newer version is supported.

In AFM and multi-cluster environment, individual clusters can be upgraded at different schedules. Access to the file system data can be preserved even though some of the clusters might still be running on an earlier version. Home or the cache cluster must be upgraded independent of the other.

During an offline upgrade, the IBM Spectrum Scale service is interrupted. For an offline upgrade, you must shut down the cluster and suspend the application workload of the cluster. During an online upgrade, IBM Spectrum Scale service is not interrupted. In an online upgrade, the system is upgraded node-by-node or failure group-by-failure group. During the upgrade, IBM Spectrum Scale runs on a subset of nodes. You can also perform offline upgrades, if you can shut down the entire cluster. An offline upgrade is similar to the online upgrade procedure. As the entire cluster is offline, it is possible to upgrade to the latest code level instead of upgrading to an intermediate level, as might be needed during an online upgrade.

Before you start an AFM upgrade, the home cluster or primary cluster for AFM-DR must be upgraded first before the cache or secondary cluster.

Before you consider an online upgrade of the home or cache, ensure that:

- The cluster is healthy and operational.
- IBM Spectrum Scale is running on all nodes that are defined in the cluster.
- All protocols that are defined on the protocol node are running.
- Ensure that the storage at the cache cluster is adequate by using `mmrepquota` command during the home upgrade. Specifically, check the storage when the cache eviction feature is enabled for the storage management.

Cache cluster - In multiple gateway environment, gateway nodes can be upgraded one by one. In these cases, filesets, which are associated with the gateway node to upgrade, are transferred to another gateway node, and any write-class operation triggers recovery feature. This recovery feature builds the queue on associated gateway node to continue processing the operations to home. Thus, cache to home are not disconnected. However, some performance degradation can be seen due to another gateway node that is working for the connect for those filesets that are hosted on the upgrade node previously. In heavy load systems, transferring the filesets to another gateway node might have a performance impact. It is advised to choose a time for such upgrades where the load on the system or the number of data transfers is minimal.

Parallel data transfers enabled with GW mappings - in multiple-gateways environment, where parallel data transfer is enabled with multiple gateways, upgrading any of these mapped nodes results in a normal data transfer path.

Note: Fileset mapping might remain intact after upgrade, depending on the afmHashing version that is being used.

When gateway nodes manage many filesets, it is recommended to perform the rolling upgrade by using start stop replication. For more information about the start and stop replication, see [“Using stop and start replication to upgrade AFM and AFM DR”](#) on page 519.

If you want the replication to continue as-is during the upgrade, you can tune the number of parallel recoveries that are run on a gateway node by using the **afmMaxParallelRecoveries** parameter. The number of filesets that are specified as the **afmMaxParallelRecoveries** value are accessed for recovery. After recoveries are complete on these specified filesets, other filesets are accessed for recovery. By default, the **afmMaxParallelRecoveries** parameter is set to 0, and the recovery process is run on all filesets. You can restrict the number of recoveries by specifying the **afmMaxParallelRecoveries** value. The restriction on recoveries conserves hardware resources.

Home cluster - Cluster Export Services (CES) provides highly available file and object services to an IBM Spectrum Scale cluster by using Network File System (NFS), Object, or Server Message Block (SMB) protocols. With CES environment, the exports at home can be seen from cache by using the CES IP addresses. These IP addresses can align to protocol nodes when the CES node that already holds the CES IP address is shut down for an upgrade. The IP addresses alignment is according to the CES IP address distribution policies. Cache might see a short disruption at the time of CES failover at home but cache filesets continue to operate.

In a non-CES or NFS Server at Home environments, home to cache disconnects until the upgrade of NFS home server is complete. In a disconnected mode, cache builds up the queue for application operations. After Home is available, these operations are pushed to Home.

Note: You must shut down the NFS server before you issue the **mmshutdown** command on home nodes. When the IBM Spectrum Scale system upgrade is complete and the node mounts the file system, you can start the NFS server.

Note: The **mmclone** command is not supported on cache for AFM and primary for AFM DR. Clones that are created at home or secondary are replicated as different files. While you upgrade to IBM Spectrum Scale 4.2.2 or later, the cache cluster must be upgraded before you consider upgrade of the home cluster.

The cache cluster and the home cluster can be upgraded by using the same method. If any of these clusters has protocol nodes, upgrade these nodes. For more information, see [“Online upgrade support for protocols and performance monitoring”](#) on page 507.

After you completed upgrade, see [“Completing the upgrade to a new level of IBM Spectrum Scale”](#) on page 572. To know upgrade support for protocols and performance monitoring, see [“Online upgrade support for protocols and performance monitoring”](#) on page 507.

Using stop and start replication to upgrade AFM and AFM DR

You can use replication stop and start method to assist an upgrade. Replication must be stopped on all filesets of a gateway node you want to upgrade.

When the replication is stopped, you can shut down the gateway node for an upgrade. Local data availability is not affected as the filesets can be accessed by other nodes. However, replication activity stops during upgrade. You must ensure that replication is restarted after the gateway node is upgraded.

To restart replication, do the following steps:

1. Plan downtime to upgrade IBM Spectrum Scale. AFM is stopped on the gateway node and GPFS file system is unmounted and shutdown.
2. Issue the following command to identify filesets that belong to the gateway node you want to upgrade:

```
mmafmctl fsname getstate |grep GatewayNode
```

3. Before upgrade, check if any messages are left in queue. Wait until all pending messages are completed.

4. Issue the following command to stop replication of the fileset:

```
mmafmctl fsname stop -j filesetname
```

5. Before you upgrade the gateway node, issue the following command to verify that the replication stopped:

```
mmafmctl fsname getstate
```

6. Unmount the file system and shut down GPFS on the gateway node.
7. Upgrade the gateway node.
8. Issue **mmstartup** to start the gateway node daemon.
9. Check the node state by issuing **mmgetstate**.
10. After the node is active, mount the file systems and issue the following command:

```
mmafmctl fsname start -j filesetname
```

Recovery is initiated and replication starts.

Note: Synchronization of data between home and cache stops during the upgrade process. Recovery is run after you start replication to ensure that data at home and cache is synchronized.

Upgrading object packages

Use these steps to upgrade IBM Spectrum Scale for object storage packages from 4.2.x or later to 5.1.x.



Attention:

- All protocol nodes must be upgraded to the newer code level along with base GPFS.
- The object storage packages on all protocol nodes must be upgraded to the same level at the same time.
- Online upgrade of the object protocol is not supported because of changes that make it incompatible with servers that are running on the previous level. The upgrade must be scheduled at a time when downtime is appropriate.
- If you are upgrading from an IBM Spectrum Scale version earlier than 5.1.x and the node operating system version is earlier than Red Hat Enterprise Linux 8.x, the nodes need to be upgraded to Red Hat Enterprise Linux 8.x during the upgrade. For more information, see [“Instructions for removing object protocol packages when upgrading protocol nodes to Red Hat Enterprise Linux 8.x”](#) on page 585.
- The object protocol is not supported in IBM Spectrum Scale 5.1.0.0. If you want to deploy object, upgrade to the IBM Spectrum Scale 5.1.0.1 or a later release.

1. From one of the protocol nodes, run the following command to stop all object services on all protocol nodes:

```
mmces service stop obj -a
```

2. On all protocol nodes, update the spectrum-scale-object package to the newer version as follows.

- Use the **yum** command to update spectrum-scale-object on Red Hat Enterprise Linux.
 - a. If needed, set up a yum repository for object packages by creating a new file in /etc/yum.repos.d, such as /etc/yum.repos.d/gpfs_object.repo, with the following content:

```
[gpfs_object]
name=gpfs_object
baseurl=file:///usr/lpp/mmfs/VERSION/object_rpms/rhel8
enabled=1
gpgcheck=0
```


Make sure that the path in the `baseurl` property corresponds to the path to the newer object packages on the system.

- b. The Object protocol in IBM Spectrum Scale 5.1.2.0 and earlier 5.1.x releases requires OpenStack 16 repositories to be available on all protocol nodes to satisfy the necessary dependencies. For information on how to set up these repositories, see [“OpenStack repository configuration required by the object protocol”](#) on page 319.
- c. After the repository is set up, run the following command to update the `spectrum-scale-object` package:

```
yum upgrade -y spectrum-scale-object
```

- d. Repeat these steps on each protocol node to update the `spectrum-scale-object` package on all protocol nodes to the newer version.
- e. If a repository file in `/etc/yum.repos.d` is created in a preceding step, remove the file after the upgrade completes.

Migrate the existing object protocol configuration to be consistent with the level of installed packages after you upgrade all protocol nodes to the newer version of the `spectrum-scale-object` package.

Note: Use the following command to find the list of configured protocol nodes:

```
mm1snode -N CESNodes
```

3. From one of the protocol nodes, run the following command:

```
mmobj config manage --version-sync
```

This command needs to be issued only one time in the cluster to update the configuration on all protocol nodes.

4. Run the following command to start object protocol-related services on all protocol nodes:

```
mmces service start obj -a
```

Related tasks

[“Upgrading object packages to IBM Spectrum Scale 5.1.x or later by using the installation toolkit”](#) on page 561

Upgrading SMB packages

Use these steps to upgrade IBM Spectrum Scale SMB packages.



Attention: All protocols must also be upgraded to the newer code level along with base GPFS.

SMB package update in an IBM Spectrum Scale cluster is done in two phases to minimize the downtime of the SMB service. The protocol nodes in the cluster are divided in two halves and SMB is updated on each half one by one. In the following steps, *NodeList1* is the comma-separated list of nodes in the first half of the cluster and *NodeList2* is the comma-separated list of nodes in the second half of the cluster.

Note: All protocol nodes that are running the SMB service must have the same version of `gpfs.smb` installed at any time. A brief outage of the SMB service is required to upgrade `gpfs.smb` to the newer version across all protocol nodes. The procedure that is outlined here is intended to reduce the outage to a minimum.

1. On the first half of the protocol nodes, do the following steps.
 - a) Issue the following command to suspend the nodes and stop all the protocol services that are running on these nodes.

```
mmces node suspend -N NodeList1 --stop
```

Note: Suspending nodes triggers IP address reassignment and client failover. For more information, see [“SMB fail-over scenarios and upgrade”](#) on page 306.

- b) On each node in the first half of protocol nodes, issue one of the following commands, depending on the operating system, to upgrade the SMB package.

- Red Hat Enterprise Linux or SLES

```
rpm -Uvh gpfs.smb*.rpm
```

The name of the IBM Spectrum Scale SMB package for SLES is in this format:
`gpfs.smb_version_string.sles15.architecture.rpm`.

Note: After you upgrade the operating system on the protocol nodes from SLES 12, you might need to uninstall any obsolete packages.

- Ubuntu

```
apt install ./gpfs.smb*.deb
```

The name of the IBM Spectrum Scale SMB package for Ubuntu 20.04 LTS is in this format:
`gpfs.smb_version_string~focal_architecture.deb`.

Note: After you upgrade the operating system on the protocol nodes from Ubuntu 16.04 or Ubuntu 18.04, uninstall any obsolete packages by issuing the following command on each of these nodes.

```
apt-get autoremove
```

2. On the second half of the protocol nodes, do the following steps.

- a) Issue the following command to suspend the nodes and stop all the protocol services that are running on these nodes.

```
mmces node suspend -N NodeList2 --stop
```

Note: This command suspends the remaining protocol nodes and results in a brief outage of the SMB service till the upgraded nodes are started again in the following step. Suspending nodes triggers IP address reassignment and client failover. For more information, see [“SMB fail-over scenarios and upgrade”](#) on page 306.

3. On the first half of the protocol nodes, issue the following command to resume the nodes and start all the protocol services that are enabled on these nodes.

```
mmces node resume -N NodeList1 --start
```

For information on viewing the health status of a node, and detailed information about the health status and potential corrective actions, see *System health monitoring use cases* and *Events* in *IBM Spectrum Scale: Problem Determination Guide*.

4. On the second half of the protocol nodes, do the following steps.

- a) On each node in the second half of protocol nodes, issue one of the following commands, depending on the operating system, to upgrade the SMB package.

- Red Hat Enterprise Linux or SLES

```
rpm -Uvh gpfs.smb*.rpm
```

The name of the IBM Spectrum Scale SMB package for SLES is in this format:
`gpfs.smb_version_string.sles15.architecture.rpm`.

Note: After you upgrade the operating system on the protocol nodes from SLES 12, you might need to uninstall any obsolete packages.

- Ubuntu

```
apt install ./gpfs.smb*.deb
```

The name of the IBM Spectrum Scale SMB package for Ubuntu 20.04 LTS is in this format:
`gpfs.smb_version_string~focal_architecture.deb`.

Note: After you upgrade the operating system on the protocol nodes from Ubuntu 16.04 or Ubuntu 18.04, uninstall any obsolete packages by issuing the following command on each of these nodes.

```
apt-get autoremove
```

- b) Issue the following command to resume the nodes and start all the protocol services that are enabled on these nodes.

```
mmces node resume -N NodeList2 --start
```

For information on viewing the health status of a node, and detailed information about the health status and potential corrective actions, see *System health monitoring use cases and Events* in *IBM Spectrum Scale: Problem Determination Guide*.

Related concepts

[“Protocol authentication configuration changes during upgrade” on page 563](#)

During IBM Spectrum Scale protocol nodes upgrade, do protocol authentication-related configuration depending on your authentication setup.

Upgrading the SMB package after upgrading OS

If you have upgraded the operating system on your protocol nodes, ensure that the SMB package is upgraded to the version required for the new OS.

After upgrading the operating system and rebuilding the GPFS portability layer (GPL), upgrade to the SMB package that is specific to the new OS by using [this upgrade procedure](#).

For SMB package names, see [“Manually installing the IBM Spectrum Scale software packages on Linux nodes” on page 359](#).

Upgrading NFS packages

Use these steps on each protocol node one by one to upgrade IBM Spectrum Scale NFS packages.



Attention:

- All protocols must also be upgraded to the newer code level along with base GPFS.
- After you upgrade from an earlier version to the current version, it might be necessary to unmount and mount the file system on the NFS clients to ensure continued access.
- Before upgrading from IBM Spectrum Scale 5.0.x to 5.1.x, remove the `gpfs.nfs-ganesha-selinux` package if present in the protocol node. If the `gpfs.nfs-ganesha-selinux` package is not removed, it might break the NFS Ganesha upgrade process.
- Before upgrading from IBM Spectrum Scale 5.0.x to 5.1.x on Ubuntu operating system, make sure that the upstream NFS Ganesha package is not installed or configured in the protocol nodes. If the NFS Ganesha package is installed or configured, then uninstall the package by using the **apt purge nfs-ganesha** command.

1. Suspend the node by issuing the following command.

```
mmces node suspend --stop
```

Note: Suspending nodes triggers IP address reassignment and client failover.

2. Upgrade NFS packages on the node by issuing one of the following commands, which is applicable for an operating system.

- For Red Hat Enterprise Linux and SLES, issue the following command.

```
rpm -Uvh NFS_Package_Name1 NFS_Package_Name2 ... NFS_Package_NameN
```

- For Ubuntu, issue the following command.

```
apt-get install ./NFS_Package_Name1 ./NFS_Package_Name2 ... ./NFS_Package_NameN
```

For a list of packages for the current IBM Spectrum Scale release, see [“Manually installing the IBM Spectrum Scale software packages on Linux nodes”](#) on page 359.

3. Resume the node by issuing the following command.

```
mmces node resume --start
```

For information about viewing the health status of a node, and detailed information about the health status and potential corrective actions, see *System health monitoring use cases and Events in IBM Spectrum Scale: Problem Determination Guide*.

For information about migrating from CNFS to CES NFS, see *Migration of CNFS clusters to CES clusters in IBM Spectrum Scale: Administration Guide*.

Related concepts

[“Protocol authentication configuration changes during upgrade”](#) on page 563

During IBM Spectrum Scale protocol nodes upgrade, do protocol authentication-related configuration depending on your authentication setup.

Upgrading call home

If you are upgrading from IBM Spectrum Scale releases 4.2.0 or 4.2.1 to release 5.0.x, call home related configuration changes are needed.

The call home function is upgraded with IBM Spectrum Scale upgrade. For more information, see [“Upgrading IBM Spectrum Scale nodes”](#) on page 508.

Call home configuration changes to be made while upgrading to IBM Spectrum Scale 5.0.x from IBM Spectrum Scale 4.2.1

When you upgrade from IBM Spectrum Scale 4.2.1 to IBM Spectrum Scale 5.0.0 or later, the existing group configuration will be incompatible with the new format.

Follow one of the given methods to deploy the existing group configuration after an upgrade:

- If the existing call home groups do not need to be saved, you can use the **mmcallhome group auto --force** command to automatically delete all the current groups, and create new groups with the necessary configuration.
- If you want to save the existing call home group configuration, follow the given steps:
 1. Find the current call home group configuration using the **mmcallhome group list** command.
 2. Delete all listed call home groups using the **mmcallhome group delete <groupName>** command.
 3. Add all the call home groups that were listed in step [“1”](#) on page 524, using the following command:

```
mmcallhome group add GroupName server [--node {all | ChildNode[,ChildNode...]}]
```

4. Set the new groups to track the global settings.
 - a. Run the **mmcallhome capability list**, **mmcallhome proxy list**, or **mmcallhome schedule list** commands to view the current global settings and default settings for the new groups.

The system displays output similar to this:

```
[root@g5001-21 ~]# mmcallhome capability list
group      callHomeNode      status
-----
global     ---                enabled
g1         g5001-21.localnet.com disabled
g2         g5001-22.localnet.com disabled
```

- b. For each group, execute the **mmcallhome capability**, **mmcallhome proxy**, the **mmcallhome schedule** commands to configure the system with the existing settings available in the global field.

After each change for each group, the corresponding group entry will disappear from the list command output. Therefore, the new group uses the global value for their setting. The system displays output similar to this:

```
[root@g5001-21 ~]# mmcallhome capability list
group      callHomeNode      status
-----
global     ---                enabled
g1         g5001-21.localnet.com disabled
g2         g5001-22.localnet.com disabled
```

Change the capability to enabled for the group g1. The g1 group is set to use the global value, and is not listed in the output of the **mmcallhome capability list** command anymore, as shown in the sample output:

```
[root@g5001-21 ~]# mmcallhome capability enable accept
Call home enabled has been set to true

Additional messages:
License acceptance specified on command line. Callhome enabled.
[root@g5001-21 ~]# mmcallhome capability list
group      callHomeNode      status
-----
global     ---                enabled
g2         g5001-22.localnet.com disabled
```

After this procedure is successfully completed, the **mmcallhome capability list**, **mmcallhome proxy list**, and **mmcallhome schedule list** commands must list only one global set of values.

Call home configuration changes to be made while upgrading to IBM Spectrum Scale 4.2.1 from IBM Spectrum Scale 4.2.0

The existing call home configurations will be lost while upgrading from IBM Spectrum Scale 4.2.0 to IBM Spectrum Scale 4.2.1 or later versions.

If you want to deploy the existing configuration after an upgrade, follow the given steps:

1. Query and save your existing configuration prior to the upgrade, using the following commands:

```
# mmcallhome group list
# mmcallhome capability list
# mmcallhome info list
# mmcallhome proxy list
# mmcallhome schedule list
# cat /usr/lpp/mmfs/data/callhome/etc/ENmain.properties | grep 'CustomerId|Email|CountryCode'
# cat /usr/lpp/mmfs/data/callhome/etc/ENlocation.properties | grep CustomerName
```

2. Execute the upgrade.
3. Reconfigure call home with the configuration that has been saved in step 1, using the **mmcallhome** command. For more information on how to configure call home, see the *Configuring call home to enable manual and automated data upload* section in the *IBM Spectrum Scale: Problem Determination Guide*.

Removing residual configuration files while upgrading to IBM Spectrum Scale 5.1.x

The `gpfs.callhome-ecc-client` package is not needed when upgrading to IBM Spectrum Scale version 5.1.x from an earlier release. This is because the whole Call Home feature is now contained within the base IBM Spectrum Scale package, `gpfs.base`.

After the upgrade, some residual configuration files that are not required anymore might still be present on Ubuntu. For example:

```
root@shortwing-vm1:~# apt list gpfs.callhome-ecc-client
Listing... Done
gpfs.callhome-ecc-client/now 5.0.2-3 all [residual-config]
```

Use the following command to remove these residual configuration files:

```
apt --purge remove gpfs.callhome-ecc-client
```

Manually upgrading the performance monitoring tool

Use these steps to upgrade performance monitoring packages.

Prerequisites:

- Before uninstalling, make sure that you stop sensors and the collector. For more information see, *Starting and stopping the performance monitoring tool in IBM Spectrum Scale: Problem Determination Guide*.
- For information on uninstalling, see [“Uninstalling the performance monitoring tool” on page 502](#).
- For information on manual installation, see [“Manually installing the performance monitoring tool” on page 376](#).
- If federation is used for performance monitoring, the collectors participating in a federation must have the same version number.

To upgrade the performance monitoring tool, you must upgrade the `gpfs.gss.pmsensors` and `gpfs.gss.pmcollector` packages:

1. Upgrade to the new version of the performance monitoring tool by using the operating system native package upgrade mechanism as shown:
`rpm -Uvh gpfs.gss.pmsensors-<version>.<os>.<arch>.rpm.`

Note: Make sure that `pmsensors` and `pmcollector` are updated using the appropriate RPM packages for the platform.

2. Restart the sensors on the protocol nodes.
3. Restart the collector on the node that previously ran the collector.

Note: If multiple collectors are used, make sure that they all have the same version.

4. Verify that the collector and the sensor are running on the node by issuing the `systemctl status pmcollector` or `systemctl status pmsensors` command.

Note: If you use additional sensor packages such as `pmswift` or `gpfs.pm-ganesha`, upgrade these packages as well.

Reinstalling performance monitoring collector on Red Hat Enterprise Linux 8.x nodes

If performance collection functionality is enabled on an IBM Spectrum Scale node and you run a `leapp` upgrade from Red Hat Enterprise Linux 7.x to 8.x, the `el7 gpfs.gss.pmcollector` package is removed and the `el8` package needs to be reinstalled after the `leapp` upgrade.

Reinstall `gpfs.gss.pmcollector` on all Red Hat Enterprise Linux 8.x collector nodes as follows.

Note: For this procedure, it is assumed that when the `gpfs.gss.pmcollector` package was removed when `leapp upgrade` was invoked, the performance monitoring collector configuration was saved in `/opt/IBM/zimon/ZIMonCollector.cfg.rpmsave`.

1. Install the applicable `gpfs.gss.pmcollector` package.

The performance monitoring el8 package is located in the `/usr/lpp/mmfs/5.x.x.x/zimon_rpms/rhel8` directory.

2. Back up the current performance monitoring collector configuration.

```
mv /opt/IBM/zimon/ZIMonCollector.cfg /opt/IBM/zimon/ZIMonCollector.cfg.orig
```

3. Restore the earlier saved version of the performance monitoring collector configuration.

```
cp /opt/IBM/zimon/ZIMonCollector.cfg.rpmsave /opt/IBM/zimon/ZIMonCollector.cfg
```

4. Restart the collector.

```
systemctl restart pmcollector
```

For more information about the performance monitoring tool, see *Configuring the Performance Monitoring tool* in *IBM Spectrum Scale: Problem Determination Guide*.

Upgrading the performance monitoring packages after upgrading Ubuntu OS

If you have upgraded the Ubuntu operating system on your IBM Spectrum Scale nodes, additional steps are required to ensure that performance monitoring packages are upgraded to the version required for the latest version of Ubuntu.

After upgrading the operating system and rebuilding the GPFS portability layer (GPL), upgrade to the performance monitoring packages that Ubuntu version by using [this upgrade procedure](#).

For performance monitoring package names for Ubuntu, see [“Manually installing the IBM Spectrum Scale software packages on Linux nodes” on page 359](#).

Manually upgrading pmswift

To upgrade `pmswift` you can either uninstall and reinstall the `pmswift rpm` or use the native `rpm` upgrade command: `rpm -Uvh pmswift-version-release.noarch.rpm`. After upgrading, restart the service using the following command: `systemctl restart pmswiftd.service`.

Uninstall `pmswift-version-release.noarch.rpm`

1. Stop the `pmsensors.service` using the following command:

```
systemctl stop pmsensors.service
```

2. If uninstalling `pmswift-4.1.1-4` or later, stop the `pmswiftd.service` using the following command:

```
systemctl stop pmswiftd.service
```

If uninstalling `pmswift-4.1.1-3`, stop the `pmprovider.service` using the following command:

```
systemctl stop pmprovider.service
```

3. Uninstall the `pmswift rpm` using the following command:

```
rpm -evh --nodeps pmswift
```

If you are uninstalling `pmswift-4.1.1-3`, it should edit the Object configuration files for all Object servers and remove the entries created at the time of installation. The Object configuration files in `/etc/swift/` directory are:

- account - *.conf
- container - *.conf
- object - *.conf
- proxy - *.conf

This should also edit the sensors configuration file, `/opt/IBM/zimon/ZIMonSensors.cfg`, to remove the Object related sensors entries created at the time of installation. If you are uninstalling pmswift-4.1.1-4 or later these files will be left alone.

4. Ensure that following directories/files are removed. If they are not removed, you can, remove them manually.
 - a. `/usr/local/swiftmon` directory or `/usr/local/pmswift` directory
 - b. `/var/log/swiftmon` directory or `/var/log/pmswift` directory
 - c. `/var/run/swiftmon` directory or `/var/run/pmswift.pid` file
 - d. For pmswift-4.1.1-4 and later remove `/etc/rc.d/init.d/pmswift` file and for pmswift-4.1.1-3 remove `/etc/rc.d/init.d/pmprovider` file
 - e. For pmswift-4.1.1-3 `SwiftAccount.cfg`, `SwiftContainer.cfg`, `SwiftObject.cfg` and `SwiftProxy.cfg` files from within the Performance Monitoring tool's installation directory, `/opt/IBM/zimon/`.
5. Ensure that for pmswift-4.1.1-3 the `pmprovider.service` and for pmswift-4.1.1-4 and later the `pmswftd.service` is not available anymore by running the following command:

```
systemctl daemon-reload
```

Install pmswift-version-release.noarch.rpm

1. Install the pmswift rpm using the following command:

```
rpm -ivh pmswift-version-release.noarch.rpm
```

2. Ensure that following directories/files have been created:
 - a. `/usr/local/pmswift` directory
 - b. `/var/log/pmswift` directory
 - c. `/etc/logrotate.d/pmswift` file
 - d. `/etc/rc.d/init.d/pmswftd` file
 - e. `SwiftAccount.cfg`, `SwiftContainer.cfg`, `SwiftObject.cfg` and `SwiftProxy.cfg` files in the Performance Monitoring tool's installation directory, `/opt/IBM/zimon/`.
3. Edit the Object configuration files for all Object servers that reside in CCR, using the `/usr/local/pmswift/bin/pmswift-config-swift set` command. CCR will then propagate modified configuration files to `/etc/swift/` directory on all the protocol nodes within the cluster. The modified configuration files are:
 - account - *.conf
 - container - *.conf
 - object - *.conf
 - proxy - *.conf
4. Edit the sensors configuration file information stored in the CCR using the `/usr/local/pmswift/bin/pmswift-config-zimon set` command to add the following Object related sensors entries:

```
{
    # SwiftAccount operational metrics
    name = "SwiftAccount"
    period = 1
    type = "generic"
```



```

        restrict= "cesNodes"
    },
    {
        # SwiftContainer operational metrics
        name = "SwiftContainer"
        period = 1
        type = "generic"
        restrict= "cesNodes"
    },
    {
        # SwiftObject operational metrics
        name = "SwiftObject"
        period = 1
        type = "generic"
        restrict= "cesNodes"
    },
    {
        # SwiftProxy operational metrics
        name = "SwiftProxy"
        period = 1
        type = "generic"
        restrict= "cesNodes"
    },
}

```

These entries are then automatically propagated to the ZIMonSensors.cfg file in /opt/IBM/zimon on all the nodes in the cluster.

5. Start the pmswiftd.service using the following command:

```
systemctl start pmswiftd.service
```

6. Start the pmsensors.service using the following command:

```
systemctl start pmsensors.service
```

Manually upgrading the IBM Spectrum Scale management GUI

You can upgrade the IBM Spectrum Scale management GUI to the latest version to get the latest features. You can upgrade one GUI node at a time without shutting down IBM Spectrum Scale on other nodes to ensure high availability.

Prerequisite

Ensure that you are aware of the following details before you start the upgrade process:

- The management GUI package must be upgraded after all other required IBM Spectrum Scale components such as gpfs.base, gpfs.java, or gpfs.gss.pmcollector have been successfully updated.
- If you are upgrading from IBM Spectrum Scale 5.0.0 or earlier versions and you store your LDAP, AD, or GUI SSL configuration in /opt/ibm/wlp/usr/servers/gpfsGui/server.xml then you need to back up your settings. After the upgrade, recreate your configuration with **mkldap** for LDAP or AD and with **sethttpskeystore** for GUI SSL certificates. For more information, see *Configuring external authentication for GUI users* in *IBM Spectrum Scale: Administration Guide*.
- Local users, user groups, user roles, and snapshot rules are not affected with an upgrade, because they exist in the cluster configure repository (CCR).
- Data in the postgres database is retained.
- All IBM Spectrum Scale packages must be of the same release on the GUI node. For example, do not mix the 4.2.3 GUI rpm with a 4.2.2 base rpm. However, GUI PTFs and fixes can usually be applied without having to install the corresponding PTF or fix of the base package. This method is helpful if you just want to resolve a GUI issue without changing anything on the base layer
- Release levels can be different in GUI nodes and other nodes of the cluster. However, the minimum release level of the cluster must be 4.2.0.0 or later for the GUI to function.
- The user ID scalemgmt must not be already used as the GUI requires this user ID to run IBM Spectrum Scale GUI Websphere Java process.
- The ports 80, 443, 474080, and 47443 must not be already used by other processes.

- You can directly upgrade the IBM Spectrum Scale GUI from 4.2.0.0 or later to the latest version.

Perform the following steps to manually upgrade the management GUI:

1. Upgrade the GUI package.

For upgrading the previously installed package, use **rpm -Fvh** or **rpm -Uvh** options on RHEL or SLES. The **rpm -Fvh** is used for upgrading the existing installed package and **rpm -Uvh** is used for installing the package and upgrading the package as well.

For upgrading on Ubuntu, use **dpkg -i**. By issuing the **dpkg -s gpfs.gui** command, you can check additional properties about your management GUI installation. If the status in the command output is `install ok installed` then the upgrade has been performed successfully.

- On RHEL or SLES, issue the following command:

```
rpm -Fvh gpfs.gui-5.1.5-x.noarch.rpm
```

- On Ubuntu, issue the following command:

```
dpkg -i gpfs.gui_5.1.5-x_all.deb
```

2. After installing the packages, you can configure the performance monitoring tools by using the management GUI, if it is required. For more information, see [“Enabling performance tools in management GUI”](#) on page 382.
3. If the minimum release level set for IBM Spectrum Scale is not same as the GUI version, change the release level by issuing the **mmchconfig release=LATEST** command. 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. For more information, see [“Completing the upgrade to a new level of IBM Spectrum Scale”](#) on page 572.
4. Issue the **systemctl status gpfsd** command to verify the GUI service status.
5. Issue the **systemctl status pmcollector** and **systemctl status pmsensors** commands to verify the status of the performance tool.

Creating a CSI administrator after new installation or upgrade

The Container Storage Interface (CSI) needs an administrator user in the GUI and REST API user base to manage the CSI Driver for IBM file storage configuration. These users need to hold the CSI Administrator role. You need to assign this user role manually as no user holds this role by default. The steps that are used to create a user with this role differ depending on upgrade or a new installation scenario.

Adding a group that holds the CSI admin role after system upgrade

The CSI administrator group is created by default when you install the IBM Spectrum Scale management GUI. If you are upgrading the system from 5.0.3 or earlier, this user group does not get created by default. To create the user group with CSI administrator role, perform the following:

1. Go to **Services > GUI > Groups**.
2. Click **Create Group**. The Create User Group dialog appears.
3. Type the name of the user group, for example, *csiAdmin*, in the **User group name** field.
4. Select **CSI Administrator** from the list of user roles.
5. Click **Create** to create the user group with the CSI Administrator user role.

Assigning CSI Administrator user role after new installation or upgrade

For new installations as well as for upgrades, you must add members to the new group with the CSI Administrator role. If you create a new user, the approach is as follows:

1. Go to **Services > GUI > User**
2. Click **Create User**. The Create User dialog appears.
3. Type the name of the user in the **Name** field.

4. Select *csiAdmin* from the list of user groups.
5. Create a temporary password for the user.
6. Click **Create** to complete user creation process.

Alternatively, you can add the existing users to a group by using the **Manage Group** option that is available in the **Actions** menu.

Upgrading Cloud services

To upgrade to a newer version of Cloud services, first consider the version that you are migrating from and then consider coexistence and compatibility issues.

You can upgrade Cloud services from one version to another. When you do so, the metrics associated with Cloud services are upgraded as well. For example, in the current version, if there are 10 elements in the metrics, and when you upgrade, there might be 15 elements in the metrics. For more information, see *List of metrics for Cloud services* topic in the *IBM Spectrum Scale: Administration Guide*.

Note: Before performing any migration, ensure that you back up the configuration files. For more information, see the *Backing up the Cloud services configuration* topic in the *IBM Spectrum Scale: Administration Guide*.

Cloud services migration consists of these topics:

- [“Upgrading to Transparent cloud tiering 1.1.2 from Transparent cloud tiering 1.1.0 or 1.1.1 ” on page 531](#)
- [“Upgrading to Cloud services 1.1.2.1 from 1.1.2 ” on page 532](#)
- [“Upgrading to Cloud services 1.1.3 from 1.1.2 ” on page 532](#)
- [“Upgrading to Cloud services 1.1.3 from 1.1.0 and 1.1.1 ” on page 533](#)
- [“Upgrading to Cloud services 1.1.4 from 1.1.2.x ” on page 534](#)
- [“Upgrading to Cloud services 1.1.4 from 1.1.3 ” on page 535](#)

Upgrading to Transparent cloud tiering 1.1.2 from Transparent cloud tiering 1.1.0 or 1.1.1

This topic describes the procedure for upgrading to Transparent cloud tiering 1.1.2 from Transparent cloud tiering 1.1.0 or 1.1.1.

Ensure the following before you start the upgrade the process:

- Transparent cloud tiering service is stopped on all the server nodes by using the **mmcloudgateway service stop** command.
- IBM Spectrum Scale is at release 4.2.2. Migration might fail if your IBM Spectrum Scale release is different.
- No data migration or recall is in progress.

Perform the following steps:

1. Copy the `gpfs.tct.server-1.1.2.x86_64.rpm` file to each of the nodes that is specified as Transparent cloud tiering server nodes.
2. Run this command:

```
rpm -Uvh gpfs.tct.server-1.1.2.x86_64.rpm
```

Note: Transparent cloud tiering statistics on the IBM Spectrum Scale GUI is enabled only when the `ENABLE_MCSTORE` parameter in the `/usr/lpp/mmfs/gui/conf/gpfsgui.properties` is set to "true". During upgrade from IBM Spectrum Scale 4.2.1.1 to 4.2.2, this parameter is reset to "false", even though its original value is "true", and the Transparent cloud tiering statistics does not show up on the IBM Spectrum Scale GUI. Therefore, it is recommended to manually set the value of the property to "true" after an upgrade. To activate this change, you must restart the `gpfsgui.service`. All

other configurations will be intact after the upgrade, and you can continue to perform data migration seamlessly as before.

3. To upgrade the client packages, issue the following command:

```
rpm -Uvh gpfs.tct.client-1.1.2.1.rpm
```

4. To upgrade the Debian package, issue the following command:

```
sudo dpkg -i gpfs.tct.client*.deb
```

5. To verify the version after upgrade, issue the following command:

```
mmcloudgateway service version
```

Upgrading to Cloud services 1.1.2.1 from 1.1.2

This topic describes the procedure for upgrading to Cloud services 1.1.2.1 from 1.1.2.

Ensure the following before you start the upgrade the process:

- Transparent cloud tiering service is stopped on all the server nodes by using the **mmcloudgateway service stop** command.
- IBM Spectrum Scale is at release 4.2.2.1. Migration might fail if your IBM Spectrum Scale release is different.
- No data migration or recall is in progress.

Perform the following steps:

1. Copy the `gpfs.tct.server-1.1.2.1.x86_64.rpm` file to each of the nodes that is specified as Transparent cloud tiering server nodes.
2. Run this command:

```
rpm -Uvh gpfs.tct.server-1.1.2.1.x86_64.rpm
```

Note: Transparent cloud tiering statistics on the IBM Spectrum Scale GUI is enabled only when the `ENABLE_MCSTORE` parameter in the `/usr/lpp/mmfs/gui/conf/gpfsgui.properties` is set to "true". During upgrade from IBM Spectrum Scale 4.2.2 to 4.2.2.1, this parameter is reset to "false", even though its original value is "true", and the Transparent cloud tiering statistics does not show up on the IBM Spectrum Scale GUI. Therefore, it is recommended to manually set the value of the property to "true" after an upgrade. To activate this change, you must restart the `gpfsgui.service`. All other configurations will be intact after the upgrade, and you can continue to perform data migration seamlessly as before.

3. For upgrading the client systems, issue this command:

```
rpm -Uvh gpfs.tct.client-1.1.2.1.rpm
```

4. For upgrading the Debian package, issue this command:

```
sudo dpkg -i gpfs.tct.client-1.1.3*.deb
```

Upgrading to Cloud services 1.1.3 from 1.1.2

This topic describes the procedure for upgrading to Cloud services 1.1.3 from 1.1.2.

Ensure the following before you start the upgrade the process:

- Transparent cloud tiering service is stopped on all the server nodes by using the **mmcloudgateway service stop** command.
- IBM Spectrum Scale is at release 4.2.3. Migration might fail if your IBM Spectrum Scale release is different.
- No data migration or recall is in progress.

Perform the following steps:

1. Copy the `gpfs.tct.server-1.1.3.x86_64.rpm` file to each of the nodes that is specified as Transparent cloud tiering server nodes.
2. Run this command:

```
rpm -Uvh gpfs.tct.server-1.1.3.x86_64.rpm
```

3. To upgrade the client nodes, issue this command:

```
rpm -Uvh gpfs.tct.client-1.1.3*.rpm
```

4. To upgrade the Debian package, issue this command:

```
sudo dpkg -i gpfs.tct.client-1.1.3*.deb
```

5. Verify the version by using the **mmcloudgateway service version** command. The system displays output similar to this:

```
Cluster minReleaseLevel: 4.2.2.0
```

Node Version	Daemon node name	TCT Type	TCT Version	Equivalent Product
---	---	---	---	---
1	c350f1u1b7.pok.stglabs.ibm.com	Server	1.1.3	4.2.3

Note: This upgrade takes longer than others because of the database schema change. The larger the database of migrated files, the longer this upgrade will take.

Upgrading to Cloud services 1.1.3 from 1.1.0 and 1.1.1

This topic describes the procedure for upgrading to Cloud services 1.1.3 from 1.1.0 or 1.1.1.

Ensure the following before you start the upgrade the process:

- Transparent cloud tiering service is stopped on all the server nodes by using the **mmcloudgateway service stop** command.
- IBM Spectrum Scale is at release 4.2.3. Migration might fail if your IBM Spectrum Scale release is different.
- No data migration or recall is in progress.

Perform the following steps:

1. Copy the `gpfs.tct.server-1.1.3.x86_64.rpm` file to each of the nodes that is specified as Transparent cloud tiering server nodes.
2. Run this command:

```
rpm -Uvh gpfs.tct.server-1.1.3.x86_64.rpm
```

3. To upgrade the client nodes, issue the following command:

```
rpm -Uvh gpfs.tct.client-1.1.3*.rpm
```

4. To upgrade the Debian package, issue the following command:

```
sudo dpkg -i gpfs.tct.client-1.1.3*.deb
```

5. Verify the version by using the **mmcloudgateway service version** command. The system displays output similar to this:

```
Cluster minReleaseLevel: 4.2.2.0
```

Node Version	Daemon node name	TCT Type	TCT Version	Equivalent Product
--------------	------------------	----------	-------------	--------------------

1	c350f1u1b7.pok.stglabs.ibm.com	Server	1.1.3 4.2.3

Upgrading to Cloud services 1.1.4 from 1.1.2.x

This topic describes the procedure for upgrading to Cloud services 1.1.4 from 1.1.2.x

Ensure the following before you start the upgrade the process:

- Uninstalling the old rpm (**xpm -e** command) and then installing the new rpm (**xpm -i**) will not work. The upgrade must be done with the **xpm -Uvh** command only.
- Transparent cloud tiering service is stopped on all the server nodes by using the **mmcloudgateway service stop** command.
- IBM Spectrum Scale is at release 5.0.0. Upgrade might fail if your IBM Spectrum Scale release is different.
- File system version is upgraded to the latest (18.00 and beyond) by using the **mmchfs** command, to leverage the default transparent recall policy. If file system version is not upgraded to a given level prior to Cloud services rpm upgrade, enabling transparent recalls for a container that is created with earlier Cloud services versions will fail. If file system version is upgraded to 18.00 and beyond after upgrading the Cloud services rpms, you must restart Cloud services by using the **mmcloudgateway service** command to be able to effectuate transparent recalls for containers.
- No data migration or recall is in progress.

Perform the following steps:

1. Copy the `gpfs.tct.server-1.1.4.x86_64.rpm` file to each of the nodes that is specified as Transparent cloud tiering server nodes.
2. Run this command:

```
xpm -Uvh gpfs.tct.server-1.1.4.x86_64.rpm
```

3. To upgrade the client nodes, issue this command:

```
xpm -Uvh gpfs.tct.client-1.1.4*.rpm
```

4. To upgrade the Debian package, issue this command:

```
sudo dpkg -i gpfs.tct.client-1.1.4*.deb
```

5. Verify the version by using the **mmcloudgateway service version** command. The system displays output similar to this:

Cluster minReleaseLevel: 5.0.0.0				
Node Version	Daemon node name	TCT Type	TCT Version	Equivalent Product

1	c350f1u1b7.pok.stglabs.ibm.com	Server	1.1.4	5.0.0

Note: This upgrade takes longer than others because of the database schema change. The larger the database of migrated files, the longer this upgrade will take.

Note:

- After the upgrade, the previous sensors are no more valid and Cloud services performance monitoring will fail on the GUI. Therefore, you must remove the old sensors and populate new ones. For more information, see [“Upgrading the Cloud services sensors ” on page 538.](#)
- After the upgrade, do not use any transparent recall policy that was applied in IBM Spectrum Scale 4.2.x and earlier releases. You might experience hangs during transparent recalls if the old policies are reused. Therefore, you must remove the older policy (which was applied before upgrade using

mmchpolicy <file-system> -P Default) and enable the transparent recall for the container using the **mmcloudgateway containerpairset update** command, after the upgrade.

Upgrading to Cloud services 1.1.4 from 1.1.3

This topic describes the procedure for upgrading to Cloud services 1.1.4 from 1.1.3.

Ensure the following before you start the upgrade the process:

- Uninstalling the old rpm (**rpm -e** command) and then installing the new rpm (**rpm -i**) will not work. The upgrade must be done with the **rpm -Uvh** command only.
- Transparent cloud tiering service is stopped on all the server nodes by using the **mmcloudgateway service stop** command.
- IBM Spectrum Scale is at release 5.0.0. Migration might fail if your IBM Spectrum Scale release is different.
- File system version is upgraded to the latest (18.00 and beyond) by using the **mmchfs** command, to leverage the default transparent recall policy. If file system version is not upgraded to a given level prior to Cloud services rpm upgrade, enabling transparent recalls for a container that is created with earlier Cloud services versions will fail. If file system version is upgraded to 18.00 and beyond after upgrading the Cloud services rpms, you must restart Cloud services by using the **mmcloudgateway service** command to be able to effectuate transparent recalls for containers.
- No data migration or recall is in progress.

Perform the following steps:

1. Copy the `gpfs.tct.server-1.1.4.x86_64.rpm` file to each of the nodes that is specified as Transparent cloud tiering server nodes.
2. Run this command:

```
rpm -Uvh gpfs.tct.server-1.1.4.x86_64.rpm
```

3. To upgrade the client nodes, issue this command:

```
rpm -Uvh gpfs.tct.client-1.1.4*.rpm
```

4. To upgrade the Debian package, issue this command:

```
sudo dpkg -i gpfs.tct.client-1.1.4*.deb
```

5. To upgrade the SLES package, issue this command:

```
rpm -Uvh gpfs.tct.server-1.1.4*.rpm
```

6. Verify the version by using the **mmcloudgateway service version** command. The system displays output similar to this:

```
Cluster minReleaseLevel: 5.0.0.0
Node Daemon node name          TCT Type          TCT Version      Equivalent Product
Version
-----
1  c350f1u1b7.pok.stglabs.ibm.com Server             1.1.4            5.0.0
```

Note: This upgrade takes longer than others because of the database schema change. The larger the database of migrated files, the longer this upgrade will take.

- After the upgrade, the previous sensors are no more valid and Cloud services performance monitoring will fail on the GUI. Therefore, you must remove the old sensors and populate new ones. For more information, see [“Upgrading the Cloud services sensors” on page 538](#).
- After the upgrade, do not use any transparent recall policy that was applied in IBM Spectrum Scale 4.2.x and earlier releases. You might experience hangs during transparent recalls if the old policies

are reused. Therefore, you must remove the older policy (which was applied before upgrade using **mmchpolicy <file-system> -P Default**) and enable the transparent recall for the container using the **mmcloudgateway containerpairset update** command, after the upgrade.

After the upgrade, the previous sensors are no more valid and Cloud services performance monitoring will fail on the GUI. Therefore, you must remove the old sensors and populate new ones. For more information, see [“Upgrading the Cloud services sensors ” on page 538.](#)

Upgrading to Cloud services 1.1.5 from 1.1.4

This topic describes the procedure for upgrading to Cloud services 1.1.5 from 1.1.4.

Ensure the following before you start the upgrade the process:

- Uninstalling the old rpm (**rpm -e** command) and then installing the new rpm (**rpm -i**) will not work. The upgrade must be done with the **rpm -Uvh** command only.
- Transparent cloud tiering service is stopped on all the server nodes by using the **mmcloudgateway service stop** command.
- IBM Spectrum Scale is at release 5.0.0. Migration might fail if your IBM Spectrum Scale release is different.
- File system version is upgraded to the latest (18.00 and beyond) by using the **mmchfs** command, to leverage the default transparent recall policy. If file system version is not upgraded to a given level prior to Cloud services rpm upgrade, enabling transparent recalls for a container that is created with earlier Cloud services versions will fail. If file system version is upgraded to 18.00 and beyond after upgrading the Cloud services rpms, you must restart Cloud services by using the **mmcloudgateway service** command to be able to effectuate transparent recalls for containers.
- No data migration or recall is in progress.

Note: Ensure that you uninstall transparent recall policy as you need to use transparent recall-enabled flag in place of the transparent recall policy.

Perform the following steps:

1. Copy the `gpfs.tct.server-1.1.5.x86_64.rpm` file to each of the nodes that is specified as Transparent cloud tiering server nodes.
2. Run this command:

```
rpm -Uvh gpfs.tct.server-1.1.5.x86_64.rpm
```

3. To upgrade the client nodes, issue this command:

```
rpm -Uvh gpfs.tct.client-1.1.5*.rpm
```

4. To upgrade the Debian package, issue this command:

```
sudo dpkg -i gpfs.tct.client-1.1.5*.deb
```

5. To upgrade the SLES package, issue this command:

```
rpm -Uvh gpfs.tct.server-1.1.5*.rpm
```

6. Verify the version by using the **mmcloudgateway service version** command. The system displays output similar to this:

```
Cluster minReleaseLevel: 5.0.1

Node Daemon node name          TCT Type          TCT Version      Equivalent Product
Version
-----
---
1  c350f1u1b7.pok.stglabs.ibm.com Server             1.1.5            5.0.1
```


Note: This upgrade takes longer than others because of the database schema change. The larger the database of migrated files, the longer this upgrade will take.

After the upgrade, the previous sensors are no more valid and Cloud services performance monitoring will fail on the GUI. Therefore, you must remove the old sensors and populate new ones. For more information, see [“Upgrading the Cloud services sensors”](#) on page 538.

Upgrading to Cloud services 1.1.6 from 1.1.5

This topic describes the procedure for upgrading to Cloud services 1.1.6 from 1.1.5.

Ensure the following before you start the upgrade the process:

- Uninstalling the old rpm (**rpm -e** command) and then installing the new rpm (**rpm -i**) will not work. The upgrade must be done with the **rpm -Uvh** command only.
- Transparent cloud tiering service is stopped on all the server nodes by using the **mmcloudgateway service stop** command.
- IBM Spectrum Scale is at release 5.0.2. Migration might fail if your IBM Spectrum Scale release is different.
- File system version is upgraded to the latest (18.00 and beyond) by using the **mmchfs** command, to leverage the default transparent recall policy. If file system version is not upgraded to a given level prior to Cloud services rpm upgrade, enabling transparent recalls for a container that is created with earlier Cloud services versions will fail. If file system version is upgraded to 18.00 and beyond after upgrading the Cloud services rpms, you must restart Cloud services by using the **mmcloudgateway service** command to be able to effectuate transparent recalls for containers.
- No data migration or recall is in progress.

Perform the following steps:

1. Copy the `gpfs.tct.server-1.1.6.x86_64.rpm` file to each of the nodes that is specified as Transparent cloud tiering server nodes.
2. Run this command:

```
rpm -Uvh gpfs.tct.server-1.1.6.x86_64.rpm
```

3. To upgrade the client nodes, issue this command:

```
rpm -Uvh gpfs.tct.client-1.1.6*.rpm
```

4. To upgrade the Debian package, issue this command:

```
sudo dpkg -i gpfs.tct.client-1.1.6*.deb
```

5. To upgrade the SLES package, issue this command:

```
rpm -Uvh gpfs.tct.server-1.1.6*.rpm
```

6. Verify the version by using the **mmcloudgateway service version** command. The system displays output similar to this:

```
Cluster minReleaseLevel: 5.0.2

Node  Daemon node name          TCT Type      TCT Version    Equivalent Product
Version
-----
---
  1   c350f1u1b7.pok.stglabs.ibm.com Server         1.1.6          5.0.2
```

Note: This upgrade takes longer than others because of the database schema change. The larger the database of migrated files, the longer this upgrade will take.

After the upgrade, the previous sensors are no more valid and Cloud services performance monitoring will fail on the GUI. Therefore, you must remove the old sensors and populate new ones. For more information, see [“Upgrading the Cloud services sensors ” on page 538.](#)

Upgrading the Cloud services sensors

The sensors in earlier versions of the Cloud services are no longer valid in the 1.1.4 version. Therefore you must change the sensors after upgrading to Cloud services version 1.1.4 from either 1.1.3.x or 1.1.2.x version as described below.

You must know which sensors are configured and active.

Perform the following steps to upgrade the sensors:

1. Remove configuration of the old sensors by issuing the following commands:

- `mmpfmon config delete --sensors MCStoreGPFSSStats`
- `mmpfmon config delete --sensors MCStoreIcstoreStats`
- `mmpfmon config delete --sensors MCStoreLWEStats`

2. Remove the old sensors files from the `/opt/IBM/zimon` folder of every Cloud services node:

- `rm MCStoreGPFSSStats.cfg`
- `rm MCStoreIcstoreStats.cfg`
- `rm MCStoreLWEStats.cfg`

3. Copy the following sensor files from the `/opt/ibm/MCStore/config/` folder to the `/opt/IBM/zimon` folder:

- `TCTDebugDbStats.cfg`
- `TCTDebugLweDestroyStats.cfg`
- `TCTFsetGpfsConnectorStats.cfg`
- `TCTFsetIcstoreStats.cfg`
- `TCTFsGpfsConnectorStats.cfg`
- `TCTFsIcstoreStats.cfg`

4. On each Cloud services server node, create a new `MCStore-sensor-definition.cfg` file under `/opt/IBM/zimon/` and copy the following sensors text into it:

```
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"
  }
}

```

5. Specify the following command to register the sensors:

```
mmperfmon config add --sensors MCStore-sensor-definition.cfg
```

6. Specify the following command to verify that all the sensors are added:

```
mmperfmon config show
```

The system displays output similar to this:

```

# This file has been generated automatically and SHOULD NOT
# be edited manually. It may be overwritten at any point
# in time.

cephMon = "/opt/IBM/zimon/CephMonProxy"
cephRados = "/opt/IBM/zimon/CephRadosProxy"
colCandidates = "jupiter-vm856.pok.stglabs.ibm.com", "jupiter-vm934.pok.stglabs.ibm.com"
colRedundancy = 2
collectors = {
    host = ""
    port = "4739"
}
config = "/opt/IBM/zimon/ZIMonSensors.cfg"
ctdbstat = ""
daemonize = T
hostname = ""
ipfixinterface = "0.0.0.0"
logfile = "/var/log/zimon/ZIMonSensors.log"
loglevel = "info"
mmcmd = "/opt/IBM/zimon/MMCmdProxy"
mmdfcmd = "/opt/IBM/zimon/MMDFProxy"
mmpmon = "/opt/IBM/zimon/MmpmonSockProxy"
piddir = "/var/run"
release = "5.1.5-x"
sensors = {
    name = "CPU"
    period = 1
},
{
    name = "Load"
    period = 1
},
{
    name = "Memory"
    period = 1
},
{
    name = "Network"
    period = 1
},
{
    name = "Netstat"
    period = 10
},
{
    name = "Diskstat"
    period = 0
},
{
    name = "DiskFree"
    period = 600
}

```

```

name = "Infiniband"
period = 0

name = "GPFSDisk"
period = 0

name = "GPFSFilesystem"
period = 10

name = "GPFSNSDDisk"
period = 10
restrict = "nsdNodes"

name = "GPFSPoolIO"
period = 0

name = "GPFSVFS"
period = 10

name = "GPFSIOC"
period = 0

name = "GPFSVI064"
period = 0

name = "GPFSDDisk"
period = 10
restrict = "nsdNodes"

name = "GPFSvFLUSH"
period = 0

name = "GPFSNode"
period = 10

name = "GPFSNodeAPI"
period = 10

name = "GPFSFilesystemAPI"
period = 10

name = "GPFSLROC"
period = 0

name = "GPFSCHMS"
period = 0

name = "GPFSAFM"
period = 0

name = "GPFSAFMFS"
period = 0

name = "GPFSAFMFSET"
period = 0

name = "GPFSRPCS"
period = 10

name = "GPFSWaiters"
period = 10

```

```

    },
    {
      name = "GPFSFilesetQuota"
      period = 3600
      restrict = "jupiter-vm856.pok.stglabs.ibm.com"
    },
    {
      name = "GPFSFileset"
      period = 300
      restrict = "jupiter-vm934.pok.stglabs.ibm.com"
    },
    {
      name = "GPFSPool"
      period = 300
      restrict = "jupiter-vm934.pok.stglabs.ibm.com"
    },
    {
      name = "GPFSDiskCap"
      period = 86400
      restrict = "jupiter-vm856.pok.stglabs.ibm.com"
    },
    {
      name = "TCTDebugDbStats"
      period = 10
      type = "Generic"
    },
    {
      name = "TCTDebugLweDestroyStats"
      period = 10
      type = "Generic"
    },
    {
      name = "TCTFsetGpfsConnectorStats"
      period = 10
      type = "Generic"
    },
    {
      name = "TCTFsetIcstoreStats"
      period = 10
      type = "Generic"
    },
    {
      name = "TCTFsGpfsConnectorStats"
      period = 10
      type = "Generic"
    },
    {
      name = "TCTFsIcstoreStats"
      period = 10
      type = "Generic"
    }
  }
  smbstat = ""

```

7. Specify the following commands to restart your pmsensors and pmcollector on the nodes:

- `service pmsensors restart`
- `service pmcollector restart`

8. Verify the status of your pmsensors and pmcollector on the nodes:

- `service pmsensors status`
- `service pmcollector status`

Upgrading to IBM Cloud Object Storage software level 3.7.2 and above

If you use IBM Cloud Object Storage software level system (3.7.x and below) as the cloud storage tier for some time and then upgrade to IBM Cloud Object Storage software level 3.7.2 or above, you must perform certain configurations. You must perform these configurations because of a change in computation of the entity tag (ETag) in IBM Cloud Object Storage software level 3.7.2.

Do the following steps to upgrade to IBM Cloud Object Storage software level 3.7.2 and above:

1. To display the configured cloud account, run this command: **mmcloudgateway account list**
2. Edit /var/MCStore/.mcstore_settings and change the configuration property "<cloudname>.provider=cleversafe" to "<cloudname>.provider=cleversafe-new".
3. Restart the Transparent cloud tiering service by running this command: **mmcloudgateway service restart**

Upgrade paths and commands for file audit logging and clustered watch folder

Use the following information to upgrade file audit logging and clustered watch folder in IBM Spectrum Scale.

File audit logging base functionality was introduced in Spectrum Scale 5.0.0.

Clustered watch folder base functionality was introduced in Spectrum Scale 5.0.3.

Upgrade commands for file audit logging and clustered watch folder

Table 45. Upgrade commands for file audit logging and clustered watch folder

IBM Spectrum Scale	To: 5.1.0	To: 5.1.1	To: 5.1.2	To: 5.1.3	To: 5.1.4	To: 5.1.5
From: 5.0.X	mmmsgqueue config -- remove-msgqueue	mmmsgqueue config -- remove-msgqueue	mmmsgqueue config -- remove-msgqueue	mmmsgqueue config -- remove-msgqueue	mmmsgqueue config -- remove-msgqueue	mmmsgqueue config -- remove-msgqueue
From 5.1.0		mmmsgqueue config -- remove-msgqueue	mmmsgqueue config -- remove-msgqueue	mmmsgqueue config -- remove-msgqueue	mmmsgqueue config -- remove-msgqueue	mmmsgqueue config -- remove-msgqueue
From 5.1.1			mmmsgqueue config -- remove-msgqueue	mmmsgqueue config -- remove-msgqueue	mmmsgqueue config -- remove-msgqueue	mmmsgqueue config -- remove-msgqueue
From 5.1.2			X	mmmsgqueue config -- remove-msgqueue	mmmsgqueue config -- remove-msgqueue	mmmsgqueue config -- remove-msgqueue
From 5.1.3				X	mmmsgqueue config -- remove-msgqueue	mmmsgqueue config -- remove-msgqueue
From 5.1.4					X	mmmsgqueue config -- remove-msgqueue

Note: X represents no commands need to be run during the upgrade.

In IBM Spectrum Scale 5.1.0, there is a large architecture change that removes the dependency for the message queue and the gpfs.kafka package. The **mmmsgqueue config --remove-msgqueue** command automatically upgrades all components that are needed and disables then re-enables all of the existing healthy audits and watches. When you run this command, there is a small outage where no events are generated. The prerequisites to run this command are as follows:

- Run **mmchconfig release=LATEST** first.
- All file system versions where file audit logging or clustered watch folder are enabled or are planned to be enabled must set to at least IBM Spectrum Scale 5.1.0 or later. See **mmfsfs <fs> -V** to query the file system version.
- When upgrading file audit logging and clustered watch folder in remote cluster environments, you must ensure the following:
 - For file audit logging, ensure that your system meets the requirements for using file audit logging with remotely mounted file systems. For more information, see [“Requirements for using file audit logging with remotely mounted file systems”](#) on page 490.
 - To avoid version conflicts, it is required that you upgrade the accessing clusters before upgrading the owning cluster. Unexpected errors might occur on the remotely mounted file systems with file audit logging enabled if the remote cluster is at IBM Spectrum Scale 5.0.1 or earlier and the owning cluster was upgraded first.

Note: For more information about issues that might occur during an upgrade, see *File audit logging issues* in the *IBM Spectrum Scale: Problem Determination Guide*.

If file audit logging or clustered watch folder has never been enabled or your cluster was installed at 5.1.2 or later, the **mmmsgqueue config --remove-msgqueue** command does not need to run.

If you are upgrading from 5.0.5.X with **msgqueue** enabled, you must disable file audit logging and clustered watch folder and then run **mmmsgqueue config --remove** prior to the upgrade. File audit logging and clustered watch folder can be enabled after the upgrade completes.

If you are upgrading from 5.1.0.X/5.1.1.X with **msgqueue** enabled, you must run **mmmsgqueue config --remove-msgqueue** prior to the upgrade. This will re-enable file audit logging and clustered watch folder automatically.

After a full cluster upgrade to 5.1.3 and later, regardless of file system level, the **mmrestorefs** command will not restore files located in the `.audit_log` fileset or configuration fileset. The current configuration will not be overwritten and audit records will not be removed or restored.

Upgrading IBM Spectrum Scale components with the installation toolkit

You can use the installation toolkit to do an upgrade of almost all IBM Spectrum Scale components from the current version of IBM Spectrum Scale to the wanted version. These components include core GPFS, NFS, SMB, object, HDFS, management GUI, file audit logging, and performance monitoring. The installation toolkit can be used to do an online upgrade, an entire cluster offline upgrade, or an upgrade with a subset of nodes offline or excluded to meet your environment needs.

For information on the upgrade phases and the flow of the upgrade procedure that is done by using the installation toolkit, see [“Upgrade process flow”](#) on page 545.

Important:

- Protocol upgrades are non-concurrent and they cause brief outages to user access. Schedule an outage window during the upgrade, which is approximately 12 minutes per protocol node.
 - Before upgrading from IBM Spectrum Scale 5.0.x to 5.1.x, remove the `gpfs.nfs-ganesha-selinux` package if it is present in the protocol node. If the `gpfs.nfs-ganesha-selinux` package is not removed, it might break the NFS Ganesha upgrade process.

- Before upgrading from IBM Spectrum Scale 5.0.x to 5.1.x on Ubuntu operating system, make sure that the upstream NFS Ganesha package is not installed or configured in the protocol nodes. If the NFS Ganesha package is installed or configured, uninstall the package by using the **apt purge nfs-ganesha** command.
- For NFS upgrade, quiesce all I/O during the upgrade. During the NFS upgrade, I/O pauses occur and depending upon the client, NFS mounts and disconnects.
- SMB requires quiescing all I/O during the upgrade. Due to the SMB clustering functions, differing SMB levels cannot co-exist within a cluster at the same time. This restriction requires a full outage of SMB during the upgrade.
- For object upgrade, quiesce all I/O during the upgrade. The object service is down or interrupted at multiple times during the upgrade process. Clients might experience errors or they might be unable to connect during this time. They must retry as needed.
- Before you proceed with the IBM Spectrum Scale upgrade, ensure that Transparent cloud tiering is not active on the cluster. If it is active, shut down Transparent cloud tiering services on all nodes by issuing the **mmcloudgateway service stop** command. If Transparent cloud tiering is active, the IBM Spectrum Scale upgrade might fail.
- If you are upgrading from IBM Spectrum Scale 5.0.0 or earlier versions and you store your LDAP, AD, or GUI SSL configuration in `/opt/ibm/wlp/usr/servers/gpfsgui/server.xml` then you need to back up your settings. After the upgrade, recreate your configuration with **mkldap** for LDAP or AD and with **sethttpskeystore** for GUI SSL certificates. For more information, see *Configuring external authentication for GUI users* in *IBM Spectrum Scale: Administration Guide*.

Before you proceed with upgrade by using the installation toolkit, ensure that you do the following steps:

- The prerequisite steps for using the installation toolkit are done. Doing these prerequisite steps are important if you are using the installation toolkit to upgrade a cluster that was created manually. For more information, see [“Preparing to use the installation toolkit” on page 404](#).
- Review limitations of the installation toolkit related to upgrade. For more information, see [“Limitations of the installation toolkit” on page 394](#).

For doing an upgrade by using the installation toolkit, a key requirement is the cluster definition file that you can create or update by using one of the following ways.

- Use the **./spectrumscale config populate** command to retrieve the cluster configuration and populate it in the cluster definition file.
- Manually enter the configuration into the cluster definition file by using the **./spectrumscale** command options.

For information about obtaining the current cluster configuration, see [“Performing online upgrade by using the installation toolkit” on page 552](#).

Important: You can enable the prompt to administrators to shut down their workloads before starting the upgrade by using the following command.

```
./spectrumscale upgrade config workloadprompt -N Node1,Node2,...
```

The prompt is displayed on every specified node. You can also enable workload prompt on all nodes in the cluster definition file by using the **-N all** option of the command.

Upgrading by using the installation toolkit

- [“Performing online upgrade by using the installation toolkit” on page 552](#).
- [“Performing offline upgrade or excluding nodes from upgrade by using installation toolkit” on page 554](#).
- [“Upgrading IBM Spectrum Scale on IBM Spectrum Archive Enterprise Edition \(EE\) nodes by using the installation toolkit” on page 562](#).

For information on upgrading HDFS, see *Upgrading CES HDFS Transparency* in *IBM Spectrum Scale: Big data and analytics support*.

Upgrading nodes that have efix installed

You can use the installation toolkit to upgrade nodes that have an efix installed.

Installing new functions after upgrade

You can use the installation toolkit to install new functions that are available in the new version that you upgraded to and that are supported by the installation toolkit. Use the following steps to do so:

1. Enable the functions in the cluster definition file by using the installation toolkit.
2. Issue the `./spectrumscale install` or the `./spectrumscale deploy` command.

Failure recovery

In case the upgrade does not complete successfully, certain steps need to be done to identify the issue and take appropriate action to recover from the upgrade failure. For more information, see *Upgrade issues* and *Upgrade recovery* in *IBM Spectrum Scale: Problem Determination Guide*.

Upgrade process flow

The upgrade procedure by using the installation toolkit is done in several phases. If you are doing manual upgrade, you can use this process flow as a reference.

Upgrade phases

- [Upgrade precheck](#)
- “Phase 1” on [page 546](#): Upgrade of all non-protocol nodes
Note: If the nodes that are being upgraded are protocol nodes, this phase is not applicable.
- “Phase 2” on [page 548](#): Upgrade of protocol nodes (except SMB and CES components)
Note: If the nodes that are being upgraded are non-protocol nodes, this phase is not applicable.
- “Phase 3” on [page 550](#): Object version sync, SMB upgrade, and CES upgrade on protocol nodes; License package upgrade on all nodes; and File audit logging message queue enabling at cluster level.
Note: If the nodes that are being upgraded are non-protocol nodes, the object version sync, SMB upgrade, and CES upgrade tasks of this phase are not applicable.
- Phase 4: CES HDFS upgrade on DataNodes and NameNodes
- [Upgrade post-check](#)

Upgrade precheck

Important: It is highly recommended to run the upgrade precheck while you are planning to upgrade, before the actual upgrade procedure. Doing this allows you adequate time to address any issues flagged during the upgrade precheck. The upgrade precheck can be done hours or days in advance and it does not have any impact on the cluster operations.

As part of the upgrade precheck, the following tasks are performed by the installation toolkit.

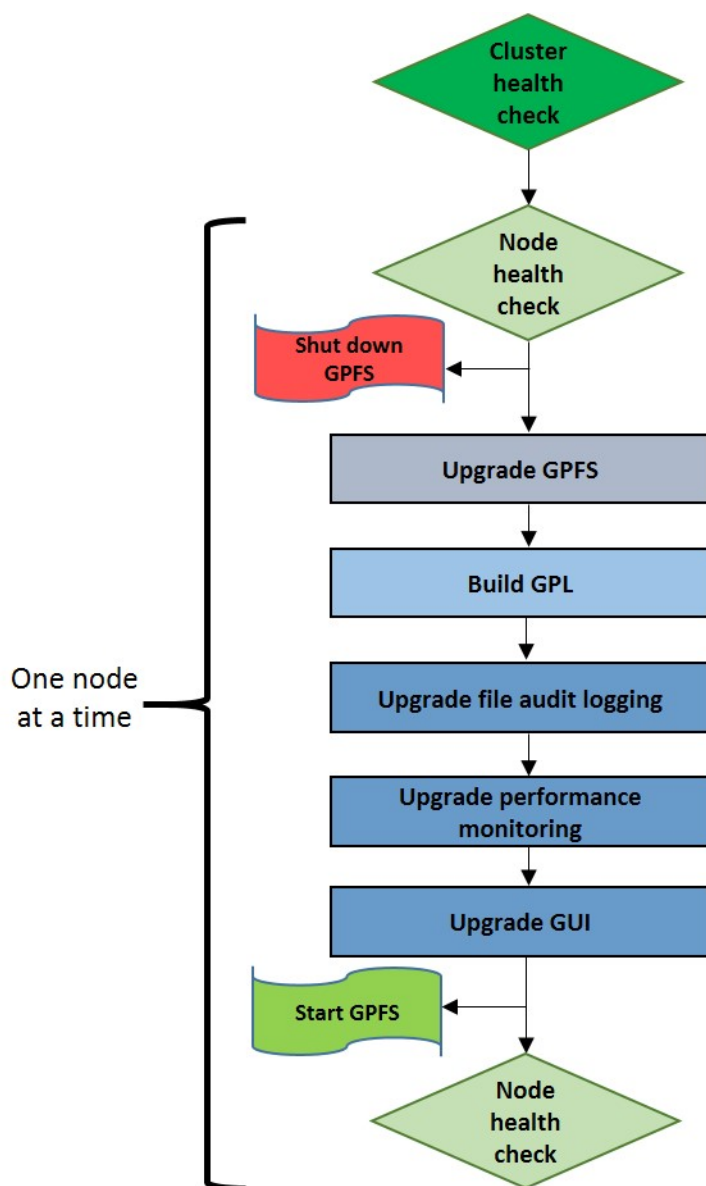
- Determine the installed IBM Spectrum Scale packages across the cluster on all nodes.
- Compare the versions of the installed packages with the versions in the repository of the packages you want to upgrade to.


In a mixed operating system cluster, the comparison is done with the package repository applicable for the operating system running on the respective nodes.


- Determine which packages need to be upgraded on each node.
- For every package that needs to be upgraded or installed, dependency check is done to ensure that all dependencies are met. The upgrade procedure throws an error and exits, if any dependency is not met.

- Check if passwordless SSH is set up.
- Check if supported OS is installed.
- Check if upgrade path is valid.

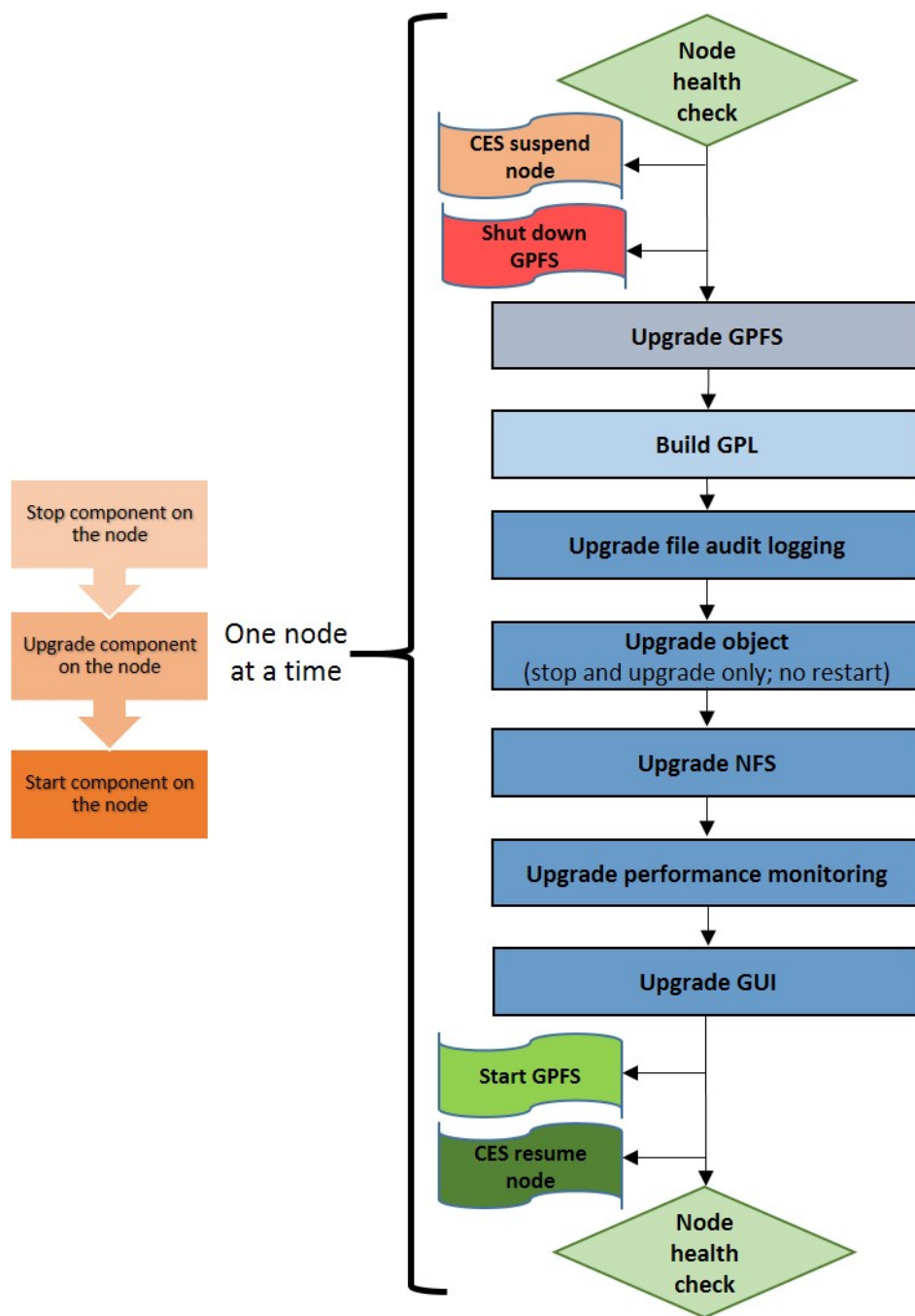
Phase 1







1	Check the cluster health.
	Perform following steps on each non-protocol node, one node at a time.
2	Check the node health.
3	Unmount file systems and shut down GPFS.  Attention: At this point, I/O from this node stops.
4	Upgrade core GPFS.
5	Build GPFS portability layer (GPL).

6	Upgrade file audit logging. Install if file audit logging is enabled in the cluster configuration but not installed.
7	Upgrade performance monitoring.
8	Upgrade GUI.
9	<p>Start up GPFS and mount the file systems.</p> <p> Attention: At this point, I/O from this node resumes.</p>
10	Check the node health.
	These steps are repeated on the remaining non-protocol nodes, one node at a time. After all of the non-protocol nodes are upgraded, phase 1 is complete.

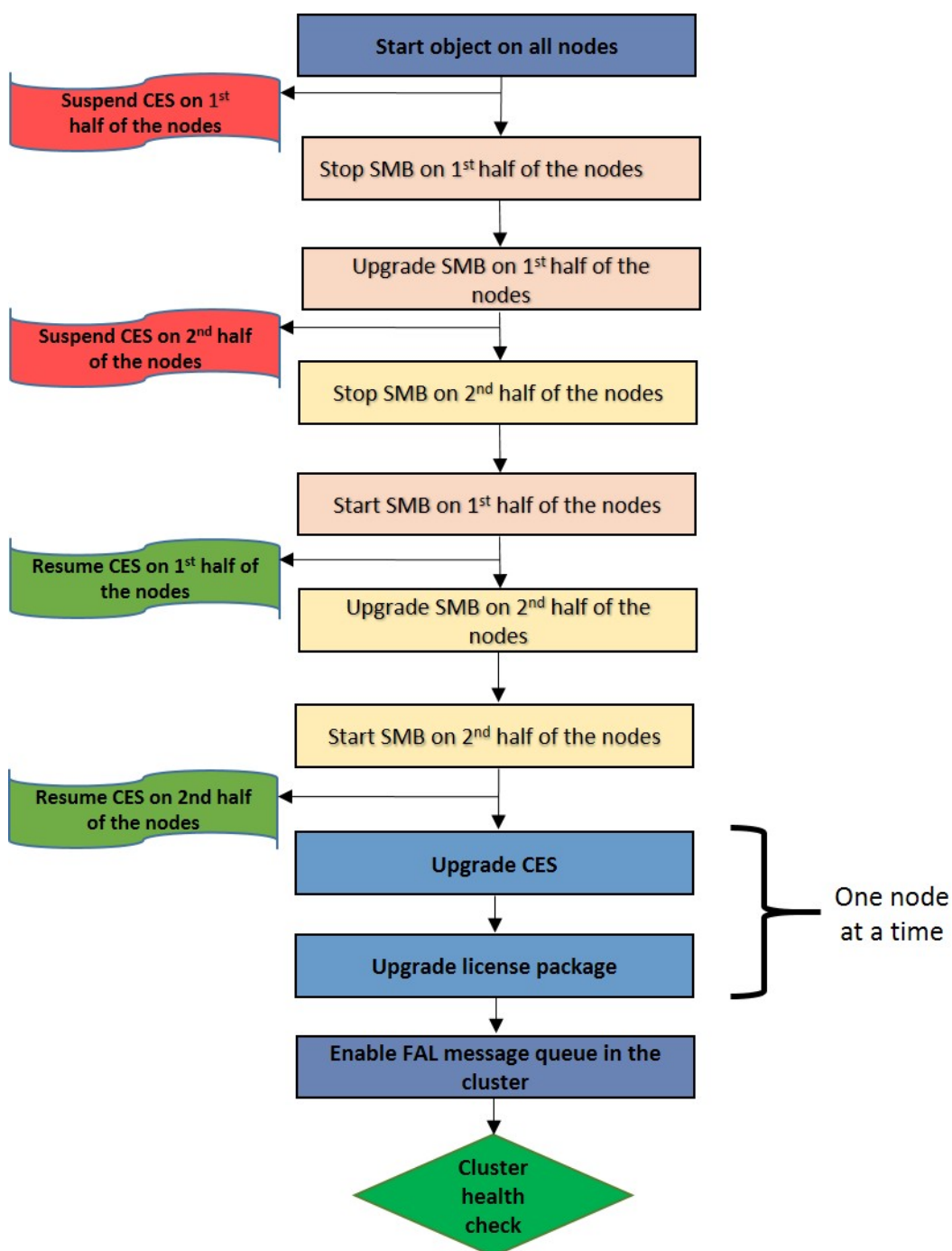
Phase 2








	Perform following steps on each protocol node, one node at a time.
1	Check the node health.
2	Suspend CES on the node.  Attention: At this point, protocol I/O from this node stops.
3	Unmount file systems and shut down GPFS.  Attention: At this point, I/O from this node stops.

4	Upgrade core GPFS.
5	Build GPFS portability layer (GPL).
6	Upgrade file audit logging. Install if file audit logging is enabled in the cluster configuration but not installed.
7	Upgrade object. Note: Object services are stopped on the node and all object packages are upgraded on the node. Object services are started on all nodes in phase 3 of the upgrade procedure.
8	Upgrade NFS.
9	Upgrade performance monitoring.
10	Upgrade GUI.
11	Start up GPFS and mount the file systems.  Attention: At this point, I/O from this node resumes.
12	Resume CES on the node.  Attention: At this point, protocol I/O from this node resumes.
13	Check the node health.
	These steps are repeated on the remaining protocol nodes, one node at a time. After all of the protocol nodes are upgraded, phase 2 is complete.

Phase 3



1	Perform object version sync from one of the protocol nodes and start object services on all protocol nodes.  Attention: At this point, object I/O resumes in the whole cluster.
2	Suspend CES on the first half of nodes.
3	Stop SMB on the 1st half of nodes.  Attention: At this point, SMB I/O on this half of nodes stops and CES IPs on these nodes are migrated to the nodes in the other half.

4	Upgrade SMB on the 1st half of nodes.
5	Suspend CES on the 2nd half of nodes.  Attention: At this point, SMB I/O stops on the whole cluster for a brief period.
6	Stop SMB on the 2nd half of nodes.
7	Start SMB on the 1st half of nodes.
8	Resume CES on the 1st half of nodes.  Attention: At this point, SMB I/O resumes on this half of nodes.
9	Upgrade SMB on the 2nd half of nodes.
10	Start SMB on the 2nd half of nodes.
11	Resume CES on the 2nd half of nodes.  Attention: At this point, SMB I/O resumes in the whole cluster.
12	Upgrade CES on all protocol nodes, one node at a time.
13	Upgrade license package on all nodes, one node at a time.
14	Enable file audit logging message queue on the cluster.
15	Check the cluster health.
	After these steps are done, the installation toolkit automatically runs upgrade post-check.

Phase 4

In this phase, as part of the CES HDFS upgrade, the `gpfs.hdfs-protocol` package is upgraded on all DataNodes and NameNodes. For more information, see *Upgrading CES HDFS Transparency* in *IBM Spectrum Scale: Big data and analytics support*.

Upgrade post-check

As part of the upgrade post-check, the following tasks are performed by the installation toolkit.

- Check the health of the nodes.
- Compare the versions of the installed packages with the versions in the package repository.
In a mixed operating system cluster, the comparison is done with the package repository applicable for the operating system running on the respective nodes.
- If any package is not upgraded, the installation toolkit prompts you to rerun the upgrade procedure.

You must complete the upgrade to the new code level to take advantage of the new functionality. For more information, see [“Completing the upgrade to a new level of IBM Spectrum Scale” on page 572](#).

Related concepts

[Performing offline upgrade or excluding nodes from upgrade by using installation toolkit](#)

[Upgrade rerun after an upgrade failure](#)

The installation toolkit supports upgrade rerun if an upgrade fails.

Related tasks

[Performing online upgrade by using the installation toolkit](#)

Use the following procedure to do an online upgrade of IBM Spectrum Scale by using the installation toolkit.

[Upgrading object packages to IBM Spectrum Scale 5.1.x or later by using the installation toolkit](#)
[Upgrading IBM Spectrum Scale on IBM Spectrum Archive Enterprise Edition \(EE\) nodes by using the installation toolkit](#)

Performing online upgrade by using the installation toolkit

Use the following procedure to do an online upgrade of IBM Spectrum Scale by using the installation toolkit.

1. Download the new IBM Spectrum Scale self-extracting package by using the substeps and then place it on the installer node.
 - a. Go to the [IBM Spectrum Scale page on Fix Central](#), select the new package, and then click **Continue**.
 - b. Choose the download option **Download using Download Director** to download the new spectrumscale package and place it in the wanted location on the install node.

Note: If you use **Download using your browser (HTTPS)**, do not click the down arrow to the left of the package name. Instead, right-click on the package name and select the **Save Link As** option. If you click the download arrow, the browser might hang.

2. Extract the new IBM Spectrum Scale self-extracting package by using the package name.
`/tmp/Spectrum_Scale_Standard-5.1.5.x_x86_64-Linux_install`

This step creates a new directory structure (`/usr/lpp/mmfs/5.1.5.x/`).

3. Accept the license agreement.

After the self-extracting package extraction is done, several next steps and instructions are displayed on the screen.

- Cluster installation and protocol deployment
- Upgrading an existing cluster by using the installation toolkit
- Adding nodes to an existing cluster by using the installation toolkit
- Adding NSDs or file systems to an existing cluster by using the installation toolkit

4. Depending on your setup, do one of the following sets of steps to obtain the current cluster configuration through the cluster definition file:

- If you are upgrading an IBM Spectrum Scale cluster that is created by using the installation toolkit, do these steps to upgrade by using the installation toolkit.

Note: If you used the installation toolkit for the initial installation and deployment, it is assumed that the prerequisites are already in place.

- a. If you want to change the installer node before the upgrade, issue the `spectrumscale setup -s InstallNodeIP` command to set up the new installer node.
- b. From the installer directory of the target version that you are upgrading to, populate the cluster definition file with the current cluster state by issuing the **`./spectrumscale config populate`** command. For more information, see [“Populating cluster definition file with current cluster state using the installation toolkit” on page 426](#) and [“Limitations of config populate option of the installation toolkit” on page 427](#).

If the command fails, update the cluster definition file with the cluster state change operations that were done after the initial installation and deployment by using the **`./spectrumscale`** commands. These cluster state change operations include:

- Node-related changes such as adding nodes
- NSD-related changes such as adding or modifying NSDs
- File system changes such as creating or modifying file systems

- Protocols-related changes such as enabling or disabling protocols
- If you are upgrading a manually created IBM Spectrum Scale cluster, do these steps to upgrade by using the installation toolkit.
 - a. Ensure that the prerequisite steps for using the installation toolkit are completed. For more information, see [“Preparing to use the installation toolkit” on page 404](#).
 - b. Configure your cluster definition file as explained in the following topics:
 - [“Setting up the installer node” on page 408](#)
 - [“Defining the cluster topology for the installation toolkit” on page 409](#)
 - c. From the installer directory of the target version that you are upgrading to, populate the cluster definition file with the current cluster state by issuing the **./spectrumscale config populate** command. For more information, see [“Populating cluster definition file with current cluster state using the installation toolkit” on page 426](#) and [“Limitations of config populate option of the installation toolkit” on page 427](#).

If the config populate command fails, update the cluster definition file with the cluster state change operations that were done after the initial installation and deployment by using the **./spectrumscale** commands. These cluster state change operations include:

- Node-related changes such as adding nodes
 - NSD-related changes such as adding or modifying NSDs
 - File system changes such as creating or modifying file systems
 - Protocols-related changes such as enabling or disabling protocols
5. Ensure that the cluster definition file is accurate and reflects the current state of your IBM Spectrum Scale cluster.

If you want to upgrade IBM Spectrum Scale on IBM Spectrum Archive Enterprise Edition (EE) nodes by using the installation toolkit, you must do some additional steps to manage IBM Spectrum Archive components. For more information, see [“Upgrading IBM Spectrum Scale on IBM Spectrum Archive Enterprise Edition \(EE\) nodes by using the installation toolkit” on page 562](#).

6. Run the upgrade precheck by using the **./spectrumscale upgrade precheck** command.

Important: It is highly recommended to run the upgrade precheck while you are planning to upgrade. Doing this step gives you adequate time to address any issues that are flagged during the upgrade precheck.

For any cluster with existing performance monitoring collectors, the upgrade prechecks might find that a reconfiguration is needed. To enable the installation toolkit to do the reconfiguration, issue the following command:

```
./spectrumscale config perfmon -r on
```

To keep your existing algorithms, issue the following command:

```
./spectrumscale config perfmon -r off
```

To do an offline upgrade based on the findings in the precheck results, do the steps in [“Designating nodes as offline in the upgrade configuration” on page 555](#).

7. Run the upgrade process by using the **./spectrumscale upgrade run** command.

Accept the warnings for system impact and interruption during the upgrade.

Note: On Ubuntu nodes, if the call home capability is enabled, after upgrading the call home packages to version 5.0.1 or later, issue the following commands from one of the nodes that is a part of a call home group.

```
mmcallhome capability disable
mmcallhome capability enable accept
```

After the upgrade is completed, the installation toolkit automatically runs the upgrade post checks. You can manually run the upgrade post checks by issuing the **./spectrumscale upgrade postcheck** command.

8. If the fresh upgrade is not successful, do an upgrade rerun as needed. For more information, see [“Upgrade rerun after an upgrade failure” on page 560](#).
9. After the upgrade process is done, complete the upgrade to the new code level to take advantage of the new functions. For more information, see [“Completing the upgrade to a new level of IBM Spectrum Scale” on page 572](#).
10. Do protocols authentication configuration that is required after you upgrade protocol nodes. For more information, see [“Protocol authentication configuration changes during upgrade” on page 563](#).

Related concepts

[Upgrade process flow](#)

[Performing offline upgrade or excluding nodes from upgrade by using installation toolkit](#)

[Upgrade rerun after an upgrade failure](#)

The installation toolkit supports upgrade rerun if an upgrade fails.

Related tasks

[Upgrading object packages to IBM Spectrum Scale 5.1.x or later by using the installation toolkit](#)

[Upgrading IBM Spectrum Scale on IBM Spectrum Archive Enterprise Edition \(EE\) nodes by using the installation toolkit](#)

Performing offline upgrade or excluding nodes from upgrade by using installation toolkit

The installation toolkit can upgrade and tolerate nodes that are in an offline state, and it can exclude some nodes from the upgrade.

Note: Online upgrades can be done to the current release (N) from the previous release (N-1). For example, to release 5.1.y (N) from release 5.0.z (N-1). However, you can use the manual offline upgrade procedure to upgrade directly to the current release (N) from a previous release (N-2) or earlier. For example, upgrading to release 5.1.x (N) from release 4.2.x (N-2) or earlier. For more information, see [“Offline upgrade with complete cluster shutdown” on page 590](#). You can also use the installation toolkit to do an offline upgrade from release 5.0.2 or later.

- [“Upgrade when nodes are unhealthy” on page 554](#)
- [“Designating nodes as offline in the upgrade configuration” on page 555](#)
- [“Upgrade when nodes are not reachable” on page 556](#)
- [“Excluding nodes from the upgrade configuration” on page 557](#)
- [“Upgrading the excluded nodes or offline designated nodes” on page 557](#)
- [“Parallel offline upgrade with the installation toolkit” on page 558](#)
- [“Upgrading mixed operating system cluster with Ubuntu nodes” on page 558](#)
- [“Upgrading mixed CPU architecture cluster” on page 559](#)
- [“Offline upgrade with the installation toolkit when OS upgrade is required” on page 559](#)
- [“Populating cluster configuration when nodes are designated as offline in the upgrade configuration” on page 559](#)
- [“Limitations of excluding nodes in the upgrade configuration” on page 560](#)

Upgrade when nodes are unhealthy

By using the IBM Spectrum Scale offline upgrade, you can upgrade your cluster even if one or more nodes are unhealthy. A node is called unhealthy when the services are down but it is reachable through ping commands.

When you designate a node as offline in the cluster configuration, during the upgrade run, the installation toolkit upgrades all installed packages. However, no attempt is made to stop or restart the respective services. You must manually restart the previously offline services by using these commands: **mmces service start** for protocol components and **mmstartup** for GPFS daemon.

For example, you try to upgrade a 5-node cluster whose nodes are node1, node2, node3, node4, and node5 (protocol node).

On upgrade precheck, you notice the following issues:

- node3 is reachable but NFS is down.
- node5 is reachable but SMB is down.
- node2 is reachable but all services, including GPFS, are down.

Here, you can do the following steps:

- Designate node3 as offline. During or after the upgrade, the installation toolkit does not restart any services, including NFS, on this node, but all installed packages, including NFS, are upgraded on node3.
- Designate node5 as offline. During or after the upgrade, the installation toolkit does not restart any services, including SMB, on this node, but all installed packages, including SMB, are upgraded on node5.
- Designate node2 as offline. All installed packages are upgraded, but none of the services are restarted.

Note:

- If you designate all nodes in a cluster as offline, then a full offline upgrade is done on all nodes. An upgrade of all installed packages is done without starting or stopping any services.
- If you try to designate a node that is already excluded as offline, then the exclude designation of the node is cleared, and the offline designation is added. For example,

```
./spectrumscale upgrade config offline -N vm1
```

```
[ INFO ] The node vm1.ibm.com was added in excluded list previously. Clearing  
this from excluded list. [ INFO ] Adding vm1.ibm.com as smb offline
```

- After an offline upgrade, you must ensure that all unhealthy services are manually started by using **mmces service start** for protocol components and **mmstartup** for GPFS.

Designating nodes as offline in the upgrade configuration

- To designate a node as offline, issue this command:

```
./spectrumscale upgrade config offline -N nodename
```

An offline upgrade is done on this node, which means that all installed packages are upgraded without restarting any services.

Important: Before you designate a node as offline, you must ensure that none of the components are active and if the node is a protocol node, then it must be suspended.

- To check the status of the GPFS daemon, issue the **mmgetstate** command.
- To stop the GPFS daemon, issue the **mmshutdown** command.
- To check the status of protocol components, issue the **mmces service list** command.
- To suspend the protocol node and stop the protocol services, issue the **mmces node suspend --stop** command.

If you are upgrading from IBM Spectrum Scale version 5.0.2.0 or earlier, issue the following commands to suspend the protocol node and stop the protocol services:

```
mmces node suspend  
mmces service stop Protocol
```

- To designate all nodes as offline and do a full offline upgrade across the cluster, issue this command:

```
./spectrumscale upgrade config offline -N node1,node2,node3...,noden
```

All installed packages are upgraded on all the nodes in the cluster, but no services are restarted on any of the nodes.

- Clearing the offline designation.

- To clear all offline designations from a specific node, issue this command:

```
./spectrumscale upgrade config offline -N nodename --clear
```

- To clear all the offline designations from all the nodes, issue this command:

```
./spectrumscale upgrade config offline --clear
```

- To clear both the offline and the exclude configuration, issue this command:

```
./spectrumscale upgrade config clear
```

- To view all configuration that is done for offline upgrade, issue this command:

```
./spectrumscale upgrade config list
```

The output of this command includes the nodes that are excluded and the nodes where the components are designated as offline. An offline upgrade is initiated based on this configuration.

For example,

```
./spectrumscale upgrade config list
```

```
[ INFO ] GPFS Node SMB NFS OBJ GPFS
[ INFO ]
[ INFO ] Phase1: Non Protocol Nodes Upgrade
[ INFO ] nsd001st001 - - -
[ INFO ] nsd002st001 - - -
[ INFO ] nsd003st001 - - -
[ INFO ] nsd004st001 - - -
[ INFO ]
[ INFO ] Phase2: Protocol Nodes Upgrade
[ INFO ] prt002st001
[ INFO ] prt003st001
[ INFO ] prt004st001
[ INFO ] prt006st001
[ INFO ] prt008st001
[ INFO ] prt009st001
[ INFO ] prt011st001
[ INFO ]
[ INFO ] Excluded Nodes : prt007st001,prt001st001,prt010st001,prt005st001
[ INFO ]
```

Upgrade when nodes are not reachable

You can exclude one or more nodes from the current upgrade run, if the nodes are unreachable or if you want to upgrade them later. When you exclude a node from the upgrade configuration, no action is done on this node during the upgrade.

Note:

- It is not recommended to exclude a subset of protocol nodes. For example, if you have three protocol nodes, then you must exclude all three nodes together. It is not recommended to exclude only a subset (1 or 2) of nodes. For example,

```
./spectrumscale upgrade config exclude -N vm1
```

```
[ INFO ] Adding node vm1.ibm.com in excluded list.
[ WARN ] Protocol nodes should all be upgraded together if possible,
since mixed versions of the code are not allowed in CES components (SMB/OBJ).
You may add the remaining protocol node(s) : vm2.ibm.com in the excluded list or clear
node(s):
```

```
vm1.ibm.com with the ./spectrumscale config exclude --clear option so that no protocol nodes are excluded.
```

- Ensure that not all admin nodes are excluded and that at least one admin node is available in the nonexcluded list. For example, if you have three admin nodes in a cluster that you want to upgrade, then you can exclude a maximum of two admin nodes only. If you have only one admin node, then it must not be excluded.

Excluding nodes from the upgrade configuration

- To exclude one or more nodes from the upgrade configuration, issue this command:

```
./spectrumscale upgrade config exclude -N node1,node2
```

This command ensures that the installation toolkit does not do any action on node1 and node2 during upgrade.

- Clearing the exclude designations.
 - To clear the exclude configuration from specific nodes, issue this command:

```
./spectrumscale upgrade config exclude -N node1,node2 --clear
```

Note: It is not recommended to clear only a subset of the protocol nodes that are designated as offline.

- To clear the exclude configuration from all nodes, issue this command:

```
./spectrumscale upgrade config exclude --clear
```

Upgrading the excluded nodes or offline designated nodes

1. Ensure that the nodes on which you want to do offline upgrade are reachable through ping commands.
2. For nodes that are designated as excluded, clear the exclude designation of the nodes in the cluster definition file by using this command:

```
./spectrumscale upgrade config offline -N node1,node2 --clear
```

3. For nodes that are designated as excluded, designate them as offline if all services are not running by using this command:

```
./spectrumscale upgrade config offline -N node1,node2
```

4. Run the upgrade procedure on the offline designated nodes by using this command:

```
./spectrumscale upgrade run
```

The installation toolkit upgrades the packages and restarts the services only for an online upgrade. For an offline upgrade, the installation toolkit only upgrades the packages that are currently installed on the offline designated nodes.

5. After the upgrade procedure is completed, do the following:
 - Restart the GPFS daemon by using the **mmstartup** command on each offline designated node.
 - If the object protocol is configured, do the post-upgrade object configuration by using the following command from one of the protocol nodes.

```
mmobj config manage --version-sync
```

- Resume the protocol node and restart the protocol services by using the **mmces node resume --start** command for every offline designated node that is a protocol node.

If you are upgrading from IBM Spectrum Scale version 5.0.2.0 or earlier, issue the following commands to resume the protocol node and start the protocol services:

```
mmces node resume
mmces service start Protocol
```

- Configure protocols authentication that is required after you upgrade protocol nodes. For more information, see [“Protocol authentication configuration changes during upgrade” on page 563](#).

Parallel offline upgrade with the installation toolkit

Use the parallel offline upgrade option to upgrade all nodes in the cluster in parallel. By using parallel offline upgrade, you can reduce the overall upgrade duration compared to doing the upgrade in a serial manner.

Note:

- You can run parallel offline upgrade only:
 - If you can shut down IBM Spectrum Scale on all nodes in the cluster.
 - If you can suspend protocol services on all protocol nodes, if applicable, in the cluster.
 - If one of the nodes in the upgrade configuration is defined as an admin node.
- To run a parallel offline upgrade, all nodes in the cluster must be designated offline in the upgrade configuration. Otherwise, all the nodes are upgraded serially. If you want to use parallel offline upgrade on a subset of nodes in the cluster, you must exclude the other nodes from the upgrade configuration by using the **./spectrumscale upgrade config exclude -N Node1, Node2,...** command. Only the nodes that are not designated as excluded in the upgrade configuration are upgraded.

Do a parallel offline upgrade as follows.

1. Shut down IBM Spectrum Scale on all nodes in the cluster.

```
# mmshutdown -a
```

2. If your cluster comprises protocol nodes, suspend protocol services on those nodes.

```
# mmces node suspend --stop -a
```

3. Designate all nodes in the cluster as offline in the upgrade configuration of the installation toolkit.

```
# ./spectrumscale upgrade config offline -N all
```

4. Run the upgrade procedure.

```
# ./spectrumscale upgrade run
```

In the 1st phase, the installation toolkit upgrades core components such as GPFS, file audit logging, performance monitoring, and GUI on all nodes in parallel. In the 2nd phase, the installation toolkit upgrades all protocol components on all protocol nodes in parallel.

Upgrading mixed operating system cluster with Ubuntu nodes

On Ubuntu nodes in a mixed operating system cluster, upgrades by using the installation toolkit are not supported. You can upgrade a mixed operating system cluster with Ubuntu nodes with the installation toolkit in two hops. You can do these hops in any order.

1. Upgrade the non-Ubuntu nodes as follows.
 - a. Extract the installation image on the non-Ubuntu installer node.
 - b. Run the config populate command.
 - c. Exclude all Ubuntu nodes from the upgrade configuration.
 - d. Run the upgrade command.

2. Upgrade the Ubuntu nodes as follows.
 - a. Extract the installation image on the Ubuntu installer node.
 - b. Run the config populate command.
 - c. Exclude all non-Ubuntu nodes from the upgrade configuration.
 - d. Run the upgrade command.

Upgrading mixed CPU architecture cluster

You can upgrade a mixed CPU architecture cluster with the installation toolkit in two hops. You can do these hops in any order.

1. Upgrade the first set of nodes that have the same CPU architecture as follows.
 - a. Extract the installation image on the installer node of the same CPU architecture.
 - b. Run the config populate command.

The cluster configuration is populated for the nodes that have the same CPU architecture as the installer node.
 - c. Run the upgrade command.
2. Upgrade the second set of nodes that have the other type of CPU architecture as follows.
 - a. Extract the installation image on the installer node for the other type of CPU architecture.
 - b. Run the config populate command.

The cluster configuration is populated for the nodes that have the same CPU architecture as the installer node.
 - c. Run the upgrade command.

Offline upgrade with the installation toolkit when OS upgrade is required

If you need to upgrade the OS during an offline upgrade of IBM Spectrum Scale with the installation toolkit, use the following high-level procedure as guidance.

1. Run the config populate command.
2. Stop the services and suspend the protocol nodes.
3. Do the OS upgrade.
4. Download and extract the IBM Spectrum Scale installation package of the version that you want to upgrade to.
5. Copy the newly generated cluster definition file to the extracted installation package location.
6. Designate the nodes that you want to upgrade as offline in the upgrade configuration.
7. Run the upgrade pre-check procedure.
8. Run the upgrade procedure.
9. Start the services and resume the protocol nodes.

Populating cluster configuration when nodes are designated as offline in the upgrade configuration

You cannot use the config populate function if one or more nodes in the cluster are designated as offline in the upgrade configuration. Use the following steps to populate the cluster configuration in scenarios in which you plan to designate one or more nodes as offline in the upgrade configuration.

1. Extract the installation image that you want to use for doing the upgrade.
2. Use the **./spectrumscale config populate** command or create a cluster definition file by using the **./spectrumscale** command.

3. Shut down one or more nodes.
4. Use the **./spectrumscale upgrade config** command to designate the nodes as offline in the upgrade configuration.

Limitations of excluding nodes in the upgrade configuration

You cannot exclude a node if SMB service is running on it. This restriction is not applicable for NFS or object. For example, when you try to exclude from the upgrade configuration the vm1 node that is running SMB, an error occurs.

```
./spectrumscale upgrade config exclude -N vm1
```

```
[ FATAL ] In order to exclude a protocol node running SMB from the current upgrade, the SMB
service must first be stopped on that node.
Please stop SMB using the mmces command and retry.
```

Related concepts

[Upgrade process flow](#)

[Upgrade rerun after an upgrade failure](#)

The installation toolkit supports upgrade rerun if an upgrade fails.

Related tasks

[Performing online upgrade by using the installation toolkit](#)

Use the following procedure to do an online upgrade of IBM Spectrum Scale by using the installation toolkit.

[Upgrading object packages to IBM Spectrum Scale 5.1.x or later by using the installation toolkit](#)

[Upgrading IBM Spectrum Scale on IBM Spectrum Archive Enterprise Edition \(EE\) nodes by using the installation toolkit](#)

Upgrade rerun after an upgrade failure

The installation toolkit supports upgrade rerun if an upgrade fails.

Do an upgrade rerun if the upgrade fails due to the following reasons:

- A component stops during the upgrade process and it is in an unhealthy state.
- Network issue during the upgrade
- Mixed mode packages

You can do a rerun by using the regular upgrade (**./spectrumscale upgrade run**) command. The installation toolkit automatically identifies whether this command is run for a fresh upgrade or for a rerun. During a rerun, the installation toolkit does an upgrade by ignoring all unrecoverable errors that you received during the upgrade.

When an upgrade rerun is done:

- The installation toolkit tries to do the upgrade even if some components become unhealthy.
- The installation toolkit displays warnings instead of unrecoverable messages if some components are in a healthy state on a node.
- The installation toolkit does not try to stop the services if they are already stopped. If the services are already running, then the installation toolkit does not try to start the services again.
- The installation toolkit tries to start the services if they are stopped.
- The installation toolkit does not try to start or stop services on nodes where the services are designated as offline and on the nodes that are designated as excluded.

Related concepts

[Upgrade process flow](#)

[Performing offline upgrade or excluding nodes from upgrade by using installation toolkit](#)

Related tasks

[Performing online upgrade by using the installation toolkit](#)

Use the following procedure to do an online upgrade of IBM Spectrum Scale by using the installation toolkit.

[Upgrading object packages to IBM Spectrum Scale 5.1.x or later by using the installation toolkit](#)

[Upgrading IBM Spectrum Scale on IBM Spectrum Archive Enterprise Edition \(EE\) nodes by using the installation toolkit](#)

Upgrading object packages to IBM Spectrum Scale 5.1.x or later by using the installation toolkit

Upgrading object packages from IBM Spectrum Scale version 5.0.x to IBM Spectrum Scale 5.1.x or later by using the installation toolkit requires certain extra steps because the object protocol is not supported on Red Hat Enterprise Linux 7.x. In IBM Spectrum Scale version 5.1.x, the object protocol is supported only on Red Hat Enterprise Linux 8.x. Therefore, object upgrade from IBM Spectrum Scale version 5.0.x involves a system upgrade from Red Hat Enterprise Linux 7.x to 8.x along with upgrade to IBM Spectrum Scale version 5.1.x.

The object protocol in IBM Spectrum Scale 5.1.2.0 and earlier 5.1.x releases requires the OpenStack installation repositories to be available on the protocol nodes to resolve the necessary dependencies. Before beginning the upgrade, ensure the systems have access to the OpenStack packages. Red Hat Enterprise Linux systems with the *Red Hat OpenStack Platform* subscription can get the packages from the subscription manager. The packages are also available through publicly accessible repositories, which are described in [OpenStack packages for RHEL and CentOS](#) on the OpenStack documentation website.

Do the following steps for upgrading object packages from IBM Spectrum Scale version 5.0.x to IBM Spectrum Scale version 5.1.x by using the installation toolkit. Because the object upgrade must be done in the offline mode, all protocol services are down during the duration of this process.

1. Stop all object services on all protocol nodes by running the following command from one of the protocol nodes.

```
mmces service stop obj -a
```

2. Clean up object-related packages on each node by using [“Instructions for removing object protocol packages when upgrading protocol nodes to Red Hat Enterprise Linux 8.x”](#) on page 585.
3. Use `leapp upgrade` to upgrade from Red Hat Enterprise Linux 7.x to 8.x on each node. For more information, see [Upgrading from RHEL 7 to RHEL 8](#) on the Red Hat documentation website.

The Object protocol in IBM Spectrum Scale 5.1.2.0 and earlier 5.1.x releases requires OpenStack 16 repositories to be available on all protocol nodes to satisfy the necessary dependencies. For information on how to set up these repositories, see [“OpenStack repository configuration required by the object protocol”](#) on page 319.

Note: The protocol nodes need to be upgraded in the offline mode.

4. Proceed with the offline upgrade by using the installation toolkit according to the instructions in [“Performing offline upgrade or excluding nodes from upgrade by using installation toolkit”](#) on page 554.

Note: When suspending the protocol nodes as part of the offline upgrade, it is expected to see error messages about missing OpenStack commands.

5. After the upgrade is completed, start IBM Spectrum Scale on the protocol nodes and resume these nodes by using the following commands.

```
mmstartup -N ProtocolNodes  
mmces node resume -N ProtocolNodes
```

6. Migrate the object protocol configuration to the latest version by running the following command from a protocol node.

```
mmobj config manage --version-sync
```

7. Restart the object protocol by running the following commands.

```
mmces service stop OBJ -a
mmces service start OBJ -a
```

Related concepts

[Upgrade process flow](#)

[Performing offline upgrade or excluding nodes from upgrade by using installation toolkit](#)

[Upgrade rerun after an upgrade failure](#)

The installation toolkit supports upgrade rerun if an upgrade fails.

Related tasks

[Performing online upgrade by using the installation toolkit](#)

Use the following procedure to do an online upgrade of IBM Spectrum Scale by using the installation toolkit.

[Upgrading IBM Spectrum Scale on IBM Spectrum Archive Enterprise Edition \(EE\) nodes by using the installation toolkit](#)

Upgrading IBM Spectrum Scale on IBM Spectrum Archive Enterprise Edition (EE) nodes by using the installation toolkit

On an IBM Spectrum Archive Enterprise Edition (EE) node, if you want to upgrade IBM Spectrum Scale, you need to do some steps before and after the upgrade to manage the IBM Spectrum Archive related components. Upgrade IBM Spectrum Scale on IBM Spectrum Archive Enterprise Edition (EE) nodes by using the installation toolkit.

Note: For latest information about IBM Spectrum Archive Enterprise Edition (EE) commands, see [IBM Spectrum Archive Enterprise Edition \(EE\) documentation](#).

1. Do the following steps that must be carried out on IBM Spectrum Archive Enterprise Edition (EE) components before an IBM Spectrum Scale upgrade.
 - a) Stop IBM Spectrum Archive Enterprise Edition (EE) by issuing the following command from one of IBM Spectrum Archive Enterprise Edition (EE) nodes.

```
eeadm cluster stop
```

- b) Deactivate failover operations by issuing the following command on all IBM Spectrum Archive Enterprise Edition (EE) nodes.

```
dsmmigfs disablefailover
```

- c) Stop the IBM Spectrum Protect for Space Management daemons by issuing the following command on all IBM Spectrum Archive Enterprise Edition (EE) nodes.

```
dsmmigfs stop
```

- d) Stop the IBM Spectrum Protect for Space Management service by issuing the following command on all IBM Spectrum Archive Enterprise Edition (EE) nodes.

```
systemctl stop hsm.service
```

After these steps, proceed with downloading, extracting, and doing other steps before an installation toolkit upgrade precheck. For more information, see [“Upgrading IBM Spectrum Scale components with the installation toolkit”](#) on page 543.

2. Run the installation toolkit upgrade precheck by issuing the following commands.

```
cd /usr/lpp/mmfs/5.1.x.x/ansible-toolkit
./spectrumscale upgrade precheck
```

Where 5.1.x.x is the release-specific directory.

This precheck fails and lists all RPMs that are failing the dependency check. Ensure that the RPMs listed are what is expected when IBM Spectrum Archive Enterprise Edition (EE) is installed.

3. Set the environment variable to bypass the installation toolkit RPM dependency check.

```
export SSFEATURE_OVERRIDE_EXT_PKG_DEPS=true
```

4. Rerun the installation toolkit upgrade precheck by issuing the following command.

```
./spectrumscale upgrade precheck
```

This precheck passes if no other problems are detected.

5. Run the installation toolkit upgrade by issuing the following command.

```
./spectrumscale upgrade run
```

6. Upgrade IBM Spectrum Archive Enterprise Edition (EE), if needed.
7. Do the following steps that must be carried out on IBM Spectrum Archive Enterprise Edition (EE) components after a successful IBM Spectrum Scale upgrade.
 - a) Start the IBM Spectrum Protect for Space Management service by issuing the following command on all IBM Spectrum Archive Enterprise Edition (EE) nodes.

```
systemctl start hsm.service
```

- b) Start the IBM Spectrum Protect for Space Management daemons by issuing the following command on all IBM Spectrum Archive Enterprise Edition (EE) nodes.

```
dsmmigfs start
```

- c) Activate failover operations by issuing the following command on all IBM Spectrum Archive Enterprise Edition (EE) nodes.

```
dsmmigfs enablefailover
```

- d) Start IBM Spectrum Archive Enterprise Edition (EE) by issuing the following command from one of IBM Spectrum Archive Enterprise Edition (EE) nodes.

```
eeadm cluster start
```

After these steps, you must complete the upgrade to new code level to take advantage of the new functions. For more information, see [“Completing the upgrade to a new level of IBM Spectrum Scale” on page 572](#).

Related concepts

[Upgrade process flow](#)

[Performing offline upgrade or excluding nodes from upgrade by using installation toolkit](#)

[Upgrade rerun after an upgrade failure](#)

The installation toolkit supports upgrade rerun if an upgrade fails.

Related tasks

[Performing online upgrade by using the installation toolkit](#)

Use the following procedure to do an online upgrade of IBM Spectrum Scale by using the installation toolkit.

[Upgrading object packages to IBM Spectrum Scale 5.1.x or later by using the installation toolkit](#)

Protocol authentication configuration changes during upgrade

During IBM Spectrum Scale protocol nodes upgrade, do protocol authentication-related configuration depending on your authentication setup.

- [“Identify the current authentication scheme configured for file protocols” on page 564](#)

- [“Upgrade authentication for file protocols set up with the LDAP-based authentication scheme” on page 564](#)
- [“Upgrade authentication for file protocols set up with the LDAP-based authentication scheme and communication with the LDAP server secured by TLS” on page 565](#)
- [“Upgrade authentication for file protocols set up with AD or LDAP file authentication with Kerberos” on page 566](#)
- [“Resolve file protocols authentication scheme configuration command failure” on page 566](#)

Identify the current authentication scheme configured for file protocols

To identify the current authentication scheme that is configured for file protocols, issue the following command.

```
# mmuserauth service list --data-access-method file
```

You identify authentication scheme that is configured for file protocols with the value of the field `FILE protocols is configured for` in the command output.

Upgrade authentication for file protocols set up with the LDAP-based authentication scheme

If file protocols are set up with the LDAP-based authentication scheme, complete the following steps. These steps are applicable for any variation of the LDAP-based authentication scheme.

Note: Issue the following commands when upgrade steps for the first protocol node in the cluster are being done. Do not repeat the following steps for the remaining protocol nodes.

1. Install the `sssd-tools` package on all the protocol nodes.
2. Obscure the password that is stored in the SSSD configuration file by doing the following steps.
 - a. Copy the current SSSD configuration file to the `/tmp` path.

```
# /bin/cp /etc/sss/sssd.conf /tmp/sssd_update.conf
```

- b. Store the secret of the LDAP user that is used to integrate with the LDAP server in current session.

```
# secret="$(/usr/lpp/mmfs/bin/mmgetconfdata -f /tmp/sssd_update.conf -s "domain/LDAPDOMAIN" -a "ldap_default_authtok")"
```

- c. Obscure the secret of the user that is used to communicate with the LDAP server by issuing the follow command.

```
# echo "${secret}" | /usr/sbin/sss_obfuscate -d LDAPDOMAIN -f /tmp/sssd_update.conf --stdin
```

- d. Clear the secret from the current session by issuing the following command.

```
# unset secret
```

- e. Publish the updated file to the CCR by issuing the follow command.

```
# /usr/lpp/mmfs/bin/mmccr fput SSSD_CONF /tmp/sssd_update.conf
```

- f. Delete the SSSD configuration file from the `/tmp` path by issuing the following command.

```
# /bin/rm /tmp/sssd_update.conf
```

3. On each protocol node, publish the updated SSSD configuration file as follows.

- a. Publish the new SSSD configuration file from IBM Spectrum Scale configuration repository to the node.

```
# /usr/lpp/mmfs/bin/mmccr fget SSSD_CONF /etc/sss/sss.conf
```

- b. Restart the SSSD service to reflect the change.

```
# systemctl restart sssd
```

4. Validate that the users from the LDAP server can be successfully resolved on all the protocol nodes.

Upgrade authentication for file protocols set up with the LDAP-based authentication scheme and communication with the LDAP server secured by TLS

If file protocols are set up with the LDAP-based authentication scheme and the communication with the LDAP server is secured by using the TLS protocol, do the following steps. These steps are applicable for any variation of the LDAP-based authentication scheme that is secured by TLS.

Note:

- You can confirm that you are using LDAP secured with TLS, if `ENABLE_SERVER_TLS` is `true` in the output of the **mmuserauth service list** command.
- Issue the following commands when upgrade steps for the first protocol node in the cluster are being done. Do not repeat the following steps for the remaining protocol nodes.

1. Fetch the configuration file from the CCR.

```
#/usr/lpp/mmfs/bin/mmccr fget LDAP_CONF /tmp/ldap_conf.from.ccr
```

2. Update the temporary file for the following changes.

- a. Delete the entry for the `TLS_CIPHER_SUITE` setting from the file.
- b. Add the following new entry to the file.

```
TLS_PROTOCOL_MIN 3.3
```

- c. Add the following new entry to the file based on an operating system (OS) of the protocol node.

If the OS of the protocol node is RHEL or SLES, add the following entry.

```
TLS_CIPHER_SUITE DEFAULT:!SSLv3:!TLSv1:!TLSv1.1:@STRENGTH
```

If the OS of the protocol node is Ubuntu, add the following entry.

```
TLS_CIPHER_SUITE NORMAL:-VERS-ALL:+VERS-TLS1.3:+VERS-TLS1.2:-AES-128-CBC:-AES-256-CBC
```

- d. Save the file.

3. Publish the updated file to the CCR by issuing the following command.

```
# /usr/lpp/mmfs/bin/mmccr fput LDAP_CONF /tmp/ldap_conf.from.ccr
```

4. Propagate the changes to all the protocol nodes of the cluster by issuing the following command.

```
# mmuserauth service check --data-access-method file --rectify
```

5. Delete the temporary file.

```
# /bin/rm /tmp/ldap_conf.from.ccr
```

Upgrade authentication for file protocols set up with AD or LDAP file authentication with Kerberos

From IBM Spectrum Scale 5.1.2, the system file `/etc/krb5.keytab` is not used for file authentication that involves Kerberos. IBM Spectrum Scale uses a custom keytab file.

If you are using AD or LDAP file authentication with Kerberos, do the following steps to upgrade the authentication configuration.

Note: You can confirm that you are using AD or LDAP file authentication with Kerberos, if either `ENABLE_NFS_KERBEROS` or `ENABLE_KERBEROS` is `true` in the output of the **mmuserauth service list** command.

1. Complete the upgrade steps on protocol nodes by using the installation toolkit or the manual upgrade procedure.
2. Run the `/usr/lpp/mmfs/bin/mmktupgrade` script.

The script creates a custom keytab file `/var/mmfs/etc/krb5_scale.keytab` and does the necessary NFS and SMB configuration changes.

3. Restart NFS and SMB on protocol nodes for the configuration changes to take effect.

Note: To prevent downtime, you can restart services on these nodes in a phased manner.

After these steps, IBM Spectrum Scale uses the custom keytab file for Kerberos authentication.

Resolve file protocols authentication scheme configuration command failure

The file protocols authentication scheme configuration command might fail with the following error.

```
mmuserauth: [E] CCR command failed: service
Failed to read value for variable LDAPMAP_DOMAINS from CCR.
mmuserauth service list: Command failed. Examine previous error messages to determine cause.
```

If this error occurs, set the `LDAPMAP_DOMAINS` variable to `none` in the cluster configuration repository (CCR) by doing the following steps from any protocol node.

1. Edit the `/tmp/authccr` file and set `LDAPMAP_DOMAINS=none`. If the `LDAPMAP_DOMAINS` entry is not present, add the entry and set it to `none`.
2. Issue this command.

```
usr/lpp/mmfs/bin/mmccr fget authccr /tmp/authccr
```



Warning: The **mmccr** command is an IBM Spectrum Scale internal component and it must be used only under the guidance of IBM Support. If the **mmccr** command is used incorrectly, the cluster can become nonoperational.

Changing the IBM Spectrum Scale product edition

You can change the IBM Spectrum Scale product edition from your currently installed product edition either by using the installation toolkit or manually.

The available product edition change paths¹ are as follows.

1. IBM Spectrum Scale Standard Edition to IBM Spectrum Scale Data Access Edition
2. IBM Spectrum Scale Standard Edition to IBM Spectrum Scale Data Management Edition
3. IBM Spectrum Scale Standard Edition to IBM Spectrum Scale Advanced Edition
4. IBM Spectrum Scale Advanced Edition to IBM Spectrum Scale Data Management Edition

The methods that can be used to change the product edition are as follows.

1. Installation toolkit (cluster online)²
2. Manual node by node (cluster online)³

3. Manual all nodes (cluster offline)

Note:

- ¹ These are the only edition changes that are supported.
- ² If you are planning to use the installation toolkit for changing the product edition, be aware that the product edition change by using the installation toolkit can be performed only when you are upgrading to a newer version of IBM Spectrum Scale. If you do not want to upgrade to a newer version of IBM Spectrum Scale, you must change the product edition manually.
- ³ You must change the whole cluster to the same product edition. If you are using the manual node by node method, ensure that the product edition is changed on each node in the cluster.

You can use one of the following sets of steps to change the IBM Spectrum Scale product edition.

- Use the following steps for changing the product edition by using the installation toolkit.
 - a) Download from IBM FixCentral the installation image of the IBM Spectrum Scale edition that you are planning to change to.
 - b) Extract the installation image.

For performing an upgrade by using the installation toolkit, a key requirement is the cluster definition file.

- c) Use the **./spectrumscale config populate** command to traverse the existing cluster and populate the cluster definition file.

For more information about obtaining the current cluster configuration, see Step 4 of [“Performing online upgrade by using the installation toolkit”](#) on page 552.

- d) Run the installation toolkit upgrade precheck procedure.

```
cd /usr/lpp/mmfs/x.x.x.x/ansible-toolkit
./spectrumscale upgrade precheck
```

Note: Replace x.x.x.x with the IBM Spectrum Scale version.

- e) Run the installation toolkit upgrade procedure.

```
cd /usr/lpp/mmfs/x.x.x.x/ansible-toolkit
./spectrumscale upgrade run
```

With these steps, the product edition is changed and the applicable packages are installed on all nodes specified in the cluster definition file.

- Use the following steps to change the product edition manually, one node at a time.

Note: In the following steps, **xpm** and **dpkg** commands are used for installing and uninstalling packages. You can also use package manager commands such as **yum**, **zypper**, and **apt-get** for installing and uninstalling packages, if the respective repository is set up in your environment.

- a) Download from IBM FixCentral the installation image of the IBM Spectrum Scale edition that you are planning to change to.
- b) Extract the installation image.
- c) Unmount all file systems on the node.

```
mmumount all
```

- d) Verify that file systems are unmounted.

```
mmismount -L -N NodeName
```

- e) Shut down GPFS on the node.

```
mmshutdown
```

- f) Uninstall the current license package using one of the following commands depending on your product edition change path and the operating system.

- IBM Spectrum Scale Standard Edition to IBM Spectrum Scale Data Access Edition or IBM Spectrum Scale Advanced Edition or IBM Spectrum Scale Data Management Edition

RHEL and SLES:

```
rpm -e License_Package_Name
```

You can obtain the license package name by using the **rpm -qa | grep gpfs.license** command.

Ubuntu:

```
dpkg -P License_Package_Name
```

You can obtain the license package name by using the **dpkg -l | grep gpfs.license** command.

- IBM Spectrum Scale Advanced Edition to IBM Spectrum Scale Data Management Edition

RHEL and SLES:

```
rpm -e License_Package_Name
```

You can obtain the license package name by using the **rpm -qa | grep gpfs.license** command.

Ubuntu:

```
dpkg -P License_Package_Name
```

You can obtain the license package name by using the **dpkg -l | grep gpfs.license** command.

- g) Install the new license package by using one of the following sets of commands depending on your product edition change path and the operating system.

- IBM Spectrum Scale Standard Edition to IBM Spectrum Scale Data Access Edition

RHEL and SLES:

```
cd /usr/lpp/mmfs/x.x.x.x/gpfs_rpms  
rpm -ivh gpfs.license.da*.rpm
```

Ubuntu:

```
cd /usr/lpp/mmfs/x.x.x.x/gpfs_debs  
dpkg -i gpfs.license.da*.deb
```

- IBM Spectrum Scale Standard Edition to IBM Spectrum Scale Data Management Edition

RHEL and SLES:

```
cd /usr/lpp/mmfs/x.x.x.x/gpfs_rpms  
rpm -ivh gpfs.license.dm*.rpm
```

Ubuntu:

```
cd /usr/lpp/mmfs/x.x.x.x/gpfs_debs  
dpkg -i gpfs.license.dm*.deb
```

- IBM Spectrum Scale Standard Edition to IBM Spectrum Scale Advanced Edition

RHEL and SLES:

```
cd /usr/lpp/mmfs/x.x.x.x/gpfs_rpms  
rpm -ivh gpfs.license.adv*.rpm
```

Ubuntu:


```
cd /usr/lpp/mmfs/x.x.x.x/gpfs_debs
dpkg -i gpfs.license.adv*.deb
```

- IBM Spectrum Scale Advanced Edition to IBM Spectrum Scale Data Management Edition
RHEL and SLES:

```
cd /usr/lpp/mmfs/x.x.x.x/gpfs_rpms
rpm -ivh gpfs.license.dm*.rpm
```

Ubuntu:

```
cd /usr/lpp/mmfs/x.x.x.x/gpfs_debs
dpkg -i gpfs.license.dm*.deb
```

- h) Start GPFS on the node.

```
mmstartup
```

- i) Repeat the preceding steps on all nodes in the cluster.

- Use the following steps to change the product edition manually, all nodes at the same time.

Note: In the following steps, **rpm** and **dpkg** commands are used for installing and uninstalling packages. You can also use package manager commands such as **yum**, **zypper**, and **apt-get** for installing and uninstalling packages, if the respective repository is set up in your environment.

- a) Download from IBM FixCentral the installation image of the IBM Spectrum Scale edition that you are planning to change to.
- b) Extract the installation image.
- c) Unmount all file systems.

```
mmumount all -a
```

- d) Verify that file systems are unmounted.

```
mmismount -L
```

- e) Shut down GPFS on all nodes.

```
mmshutdown -a
```

- f) Uninstall the current license package using one of the following commands depending on your product edition change path and the operating system.

- IBM Spectrum Scale Standard Edition to IBM Spectrum Scale Data Access Edition or IBM Spectrum Scale Advanced Edition or IBM Spectrum Scale Data Management Edition

RHEL and SLES:

```
mmdsh -N NodeList rpm -e License_Package_Name
```

You can obtain the license package name by using the **rpm -qa | grep gpfs.license** command.

Ubuntu:

```
mmdsh -N NodeList dpkg -P License_Package_Name
```

You can obtain the license package name by using the **dpkg -l | grep gpfs.license** command.

- IBM Spectrum Scale Advanced Edition to IBM Spectrum Scale Data Management Edition
RHEL and SLES:

```
mmdsh -N NodeList rpm -e License_Package_Name
```

You can obtain the license package name by using the **rpm -qa | grep gpfs.license** command.

Ubuntu:

```
mmdsh -N NodeList dpkg -P License_Package_Name
```

You can obtain the license package name by using the **dpkg -l | grep gpfs.license** command.

NodeList is a file that contains the list of nodes.

g) Use one of the following sets of commands depending on your product edition change path and the operating system.

– Install the new license package.

- IBM Spectrum Scale Standard Edition to IBM Spectrum Scale Data Access Edition

RHEL and SLES:

```
mmdsh -N NodeList rpm -ivh /usr/lpp/mmfs/x.x.x.x/gpfs_rpms/gpfs.license.da*.rpm
```

Ubuntu:

```
mmdsh -N NodeList dpkg -i /usr/lpp/mmfs/x.x.x.x/gpfs_debs/gpfs.license.da*.deb
```

- IBM Spectrum Scale Advanced Edition to IBM Spectrum Scale Data Management Edition

RHEL and SLES:

```
mmdsh -N NodeList rpm -ivh /usr/lpp/mmfs/x.x.x.x/gpfs_rpms/gpfs.license.dm*.rpm
```

Ubuntu:

```
mmdsh -N NodeList dpkg -i /usr/lpp/mmfs/x.x.x.x/gpfs_debs/gpfs.license.dm*.deb
```

– Install the new license package, and **gpfs.adv** and **gpfs.crypto** packages.

- IBM Spectrum Scale Standard Edition to IBM Spectrum Scale Data Management Edition

RHEL and SLES:

```
mmdsh -N NodeList rpm -ivh /usr/lpp/mmfs/x.x.x.x/gpfs_rpms/gpfs.license.dm*.rpm
mmdsh -N NodeList rpm -ivh /usr/lpp/mmfs/x.x.x.x/gpfs_rpms/gpfs.crypto*.rpm
mmdsh -N NodeList rpm -ivh /usr/lpp/mmfs/x.x.x.x/gpfs_rpms/gpfs.adv*.rpm
```

Ubuntu:

```
mmdsh -N NodeList dpkg -i /usr/lpp/mmfs/x.x.x.x/gpfs_debs/gpfs.license.dm*.deb
mmdsh -N NodeList dpkg -i /usr/lpp/mmfs/x.x.x.x/gpfs_debs/gpfs.crypto*.deb
mmdsh -N NodeList dpkg -i /usr/lpp/mmfs/x.x.x.x/gpfs_debs/gpfs.adv*.deb
```

- IBM Spectrum Scale Standard Edition to IBM Spectrum Scale Advanced Edition

RHEL and SLES:

```
mmdsh -N NodeList rpm -ivh /usr/lpp/mmfs/x.x.x.x/gpfs_rpms/gpfs.license.adv*.rpm
mmdsh -N NodeList rpm -ivh /usr/lpp/mmfs/x.x.x.x/gpfs_rpms/gpfs.crypto*.rpm
mmdsh -N NodeList rpm -ivh /usr/lpp/mmfs/x.x.x.x/gpfs_rpms/gpfs.adv*.rpm
```

Ubuntu:

```
mmdsh -N NodeList dpkg -i /usr/lpp/mmfs/x.x.x.x/gpfs_debs/gpfs.license.adv*.deb
mmdsh -N NodeList dpkg -i /usr/lpp/mmfs/x.x.x.x/gpfs_debs/gpfs.crypto*.deb
mmdsh -N NodeList dpkg -i /usr/lpp/mmfs/x.x.x.x/gpfs_debs/gpfs.adv*.deb
```

NodeList is a file that contains the list of nodes.

h) Start GPFS on all nodes.

```
mmstartup -a
```

Changing Express Edition to Standard Edition

Starting with release 4.2.3, IBM Spectrum Scale Express Edition is no longer available. Use this information for migrating from IBM Spectrum Scale Express Edition 4.2.2.x or earlier to IBM Spectrum Scale Standard Edition 4.2.3.x or later.

Before you begin, extract the IBM Spectrum Scale software as described in the [“Extracting the IBM Spectrum Scale software on Linux nodes”](#) on page 355 topic. For information about the location of extracted installation images, see [“Location of extracted packages”](#) on page 361.

Use one of the following sets of steps depending on the operating system and the IBM Spectrum Scale version.

Red Hat Enterprise Linux and SLES

- To migrate from 4.1.1.x or 4.2.0.x or 4.2.1.x Express Edition to 4.2.3 Standard Edition or later on Red Hat Enterprise Linux and SLES, update the existing RPMs and install new RPMs by issuing this command.

```
rpm -Uvh gpfs*rpm
```

- To migrate from 4.2.2.x Express Edition to 4.2.3 Standard Edition or later on Red Hat Enterprise Linux and SLES, use these steps.

- a) Uninstall the Express Edition license RPM by issuing the following command.

```
rpm -e gpfs.license.exp*.rpm
```

- b) Update the existing RPMs and install new RPMs by issuing the following command.

```
rpm -Uvh gpfs*rpm  
rpm -ivh gpfs.compression*.rpm  
rpm -ivh gpfs.license.std*.rpm
```

Debian and Ubuntu

- To migrate from 4.1.1.x or 4.2.0.x or 4.2.1.x Express Edition to 4.2.3 Standard Edition or later on Debian and Ubuntu, update the existing packages and install new packages by issuing the following command.

```
dpkg -i gpfs*deb
```

- To migrate from 4.2.2.x Express Edition to 4.2.3 Standard Edition or later on Debian and Ubuntu, use these steps.

- a) Uninstall the Express Edition license package by issuing the following command.

```
dpkg -r gpfs.license.exp*deb
```

- b) Update the existing packages and install new packages by issuing the following command.

```
dpkg -i gpfs*deb
```

AIX

- Depending on whether you want to be able to revert to the previous version without reinstalling or uninstalling or not, do one of the following sets of steps.
 - If you want to be able to revert to previous version without reinstalling or uninstalling, do one of following sets of steps:
 - To migrate from 4.2.0.x or 4.2.1.x Express Edition to 4.2.3 Standard Edition or later on AIX, use these steps.
 1. Update existing packages to 4.2.3 level or a later version by using the PTF images on FixCentral.

2. Install additional packages from CD images.
- To migrate from 4.2.2.x Express Edition to 4.2.3 Standard Edition on AIX, use these steps.
 1. Update existing packages to 4.2.3 level or a later version by using the PTF images on FixCentral.
 2. Install additional packages from CD images.
 3. After a successful update, uninstall the `gpfs.license.exp` package.

To revert to the previous version, do the following steps:

1. Uninstall all additional packages installed from CD images.
2. Reject the PTF images.
3. If you are reverting to 4.2.2.x and you had uninstalled the `gpfs.license.exp` package for migrating to Standard Edition, install the `gpfs.license.exp` package.
- If you can reinstall or uninstall to revert to the previous version, do one of the following sets of steps:
 - To migrate from 4.2.0.x or 4.2.1.x Express Edition to 4.2.3 Standard Edition or later on AIX, install 4.2.3 packages from CD images.
 - To migrate from 4.2.2.x Express Edition to 4.2.3 Standard Edition or later on AIX, use these steps.
 1. Install 4.2.3 or a later version packages from CD images.
 2. After a successful update, uninstall the `gpfs.license.exp` package.

Windows

- To migrate from 4.2.2.x or earlier Express Edition to Standard Edition 4.2.3 or later on Windows, use these steps.
 - a) Uninstall the current IBM Spectrum Scale packages (For example, `gpfs.base-4.x.x.0-Windows.msi`, `gpfs.gskit-8.0.x.x.msi`, and `gpfs.x.license.msi`).
 - b) Reboot the system.
 - c) Install `gpfs.ext-4.x.x.-Windows-license.msi`, `gpfs.ext-4.x.x.0-Windows.msi`, and `gpfs.gskit-8.0.x.x.msi` packages.

Completing the upgrade to a new level of IBM Spectrum Scale

It is a good idea to use the cluster for a while with the new level of IBM Spectrum Scale installed, until you are sure that you are ready to permanently upgrade the cluster to the new level.

When you are ready to permanently upgrade the cluster, follow the steps in this topic to complete the upgrade. If you decide not to complete the upgrade, you can revert to the previous level of IBM Spectrum Scale. For more information, see [“Reverting to the previous level of IBM Spectrum Scale” on page 577](#).

Before you begin this task, verify that you have upgraded all the nodes in the cluster to the latest licensed version of IBM Spectrum Scale.

When you run `mmchconfig release=LATEST` in Step 2 of these directions, you can add other parameters and their values to the command line:

```
mmchconfig release=LATEST,<parameterN=value>,<parameterN+1=value>...
```

The following table describes the options and parameters that are referred to in this topic.

Table 46. Other options and parameters with `mmchconfig release=LATEST`

Option/Parameter	Purpose	Comment
<code>--accept-empty-cipherlist-security</code>	<p>You must specify this option if you want to continue running the cluster with the lowest level of security for communications between nodes or with other clusters. That is, you want the <code>cipherList</code> attribute to remain set to <code>EMPTY</code> or undefined.</p> <p>For more information, see the topic <i>Security mode</i> in the <i>IBM Spectrum Scale: Administration Guide</i></p>	<p>It is a good idea to have some level of security for cluster communications:</p> <ul style="list-style-type: none"> Set <code>cipherList</code> to <code>AUTHONLY</code> or to a supported cipher: <pre>mmchconfig cipherList=AUTHONLY</pre> <ul style="list-style-type: none"> Do not include the parameter <code>--accept-empty-cipherlist-security</code> when you run <code>mmchconfig release=LATEST</code>. <p>For more information, see the topic <i>Security mode</i> in the <i>IBM Spectrum Scale: Administration Guide</i></p>
<code>--accept-no-compliance-to-nist-standards</code>	<p>You must specify this option if you do not want to continue running the cluster with the security transport that follows the NIST SP800-131A recommendations.</p> <p>For more information, see the topic <i>NIST compliance</i> in the <i>IBM Spectrum Scale: Administration Guide</i></p>	<p>It is recommended to run the security transport that follows the NIST SP800-131A recommendations:</p> <ul style="list-style-type: none"> Set <code>nistCompliance</code> attribute to <code>SP800-131A</code> <pre>mmchconfig nistCompliance=SP800-131A</pre> <ul style="list-style-type: none"> Set <code>nistCompliance</code> attribute to <code>SP800-131A</code> at the same time when you upgrade IBM Spectrum Scale to a new level. <pre>mmchconfig release=LATEST,nistCompliance=SP800-131A</pre>

Table 46. Other options and parameters with mmchconfig release=LATEST (continued)		
Option/Parameter	Purpose	Comment
mmfsLogTimeStampISO8601={yes no}	<p>Setting this parameter to no allows the cluster to continue running with the earlier log time stamp format.</p> <p>For more information, see the topic <i>Security mode</i> in the <i>IBM Spectrum Scale: Administration Guide</i></p>	<ul style="list-style-type: none"> Set mmfsLogTimeStampISO8601 to no if you save log information and you are not yet ready to switch to the new log time stamp format. After you complete the upgrade, you can change the log time stamp format at any time with the mmchconfig command. Omit this parameter if you are ready to switch to the new format. The default value is yes
tscCmdAllowRemoteConnections	<p>Setting this attribute to no forces the ts* commands to communicate with the local mmfsd daemon only over a UNIX domain socket (UDS), and to communicate with mmfsd daemons running on other nodes over the RPC communication framework used by IBM Spectrum Scale.</p>	<p>After a cluster is upgraded, the tscCmdAllowRemoteConnections parameter value is set to yes by default. Setting the tscCmdAllowRemoteConnections configuration to no requires a cluster minimum release level of 5.1.3 or later. System administrators can set the tscCmdAllowRemoteConnections value to no after setting the minReleaseLevel=LATEST, which is 5.1.3 or later.</p> <p>Setting tscCmdAllowRemoteConnections to no in a multi-cluster setting must be done after ensuring that the remote clusters are running Spectrum Scale version 5.1.3 or later. For more information, see <i>mmchconfig command</i> in <i>IBM Spectrum Scale: Command and Programming Reference</i>.</p>

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.

1. Verify that the SHA message digest and the cipherList configuration variable are set to valid values.

Note:

- The SHA message digest is a hash result that is generated by a cryptographic hash function.
- The cipherList variable specifies the security mode for communications among nodes in the cluster and with nodes in other clusters.

Follow these steps:

- a) Display the current values by entering the following command. The listing shows both the command and example output:

```
# mmauth show .
```

```
Cluster name: zounds.cluster (this cluster) Cipher list: (none specified)
SHA digest: (undefined) File system access: (all rw)
```

- b) If the value for the SHA digest is (undefined), follow these steps:

- i) Enter the following command to generate a public/private key pair and an SHA message digest:

```
mmauth genkey new
```

- ii) Enter `mmauth show .` again and verify that the value for SHA digest is no longer (undefined).

- c) If the value for cipherList is (none specified) or EMPTY, do one of the following actions:

- If you want a level of security in communications between nodes and with other clusters, follow these steps:

- i) Set cipherList to AUTHONLY or to a supported cipher:

```
mmauth update . -l AUTHONLY
```

- ii) Enter `mmauth show .` again and verify that the value for cipherList is no longer (none specified) or EMPTY.

- If you do not want a level of security cluster communications, let cipherList remain set to (none specified) or EMPTY.

2. Enter the following command to upgrade the cluster configuration data and enable new functionality. You can add the parameters that are described in [Table 46 on page 573](#) to the command line:

```
mmchconfig release=LATEST
```

Note: Until you run the **mmchconfig release=LATEST** command, the management GUI might not be fully operational at the new code level.

Important: If the `mmchconfig` command detects any nodes that are not available or cannot be reached, it lists the names of those nodes. If any such nodes are listed, correct the problem and run the command again until it verifies all the nodes and completes successfully.

Note: If the `mmchconfig` command fails with an error message that indicates that cipherlist is set to EMPTY, do one of the following actions:

- If you want the cluster to run with a higher security mode than EMPTY, set cipherList to AUTHONLY or to a supported cipher:

```
mmauth update . -l AUTHONLY
```

Return to the first part of Step 2 and run the `mmchconfig` command as before.

- If you want the cluster to continue with the security mode set to EMPTY, return to the first part of Step 2 and run the `mmchconfig` command with the additional parameter `--accept-empty-cipherlist-security`.

3. If you have not already done so, assign an appropriate IBM Spectrum Scale license to each of the nodes in the cluster.

See [“IBM Spectrum Scale license designation” on page 215](#) for a detailed description of the IBM Spectrum Scale license types. To see what the minimum required IBM Spectrum Scale license is for each of the nodes in the cluster, enter the following command:

```
mmllslicense -L
```

To assign an IBM Spectrum Scale server license to the nodes that require it, enter the following command:

```
mmchlicense server -N NodeList
```

To assign an IBM Spectrum Scale client license to the nodes that require it, enter:

```
mmchlicense client -N NodeList
```

4. Enable backward-compatible format changes or upgrade all file systems to the latest metadata format changes.



Attention: Before you continue with this step, it is important to understand the differences between `mmchfs -V compat` and `mmchfs -V full`:

- If you enter `mmchfs -V compat`, only changes that are backward compatible with the previous file system version are enabled. Nodes in remote clusters that are running at the previous IBM Spectrum Scale version or later will still be able to mount the file system. Nodes running a IBM Spectrum Scale version earlier than the previous version will no longer be able to mount the file system. For example, if you are upgrading from IBM Spectrum Scale 5.0.2.x to 5.1.0.x, a file system of version corresponding to IBM Spectrum Scale 5.0.2.x will remain mountable from a remote cluster running IBM Spectrum Scale 5.0.2.x or later. However, a file system of version corresponding to IBM Spectrum Scale 5.0.0.0 will remain mountable from a remote cluster running IBM Spectrum Scale 5.0.0.0 or later.

Running in the compatibility mode might prevent enablement of new functionality that relies on disk structures available only with the latest version.

- If you enter `mmchfs -V full`, all new functions that require different on-disk data structures are enabled. Nodes in remote clusters that run an older IBM Spectrum Scale version will no longer be able to mount the file system. For example, remote container native clusters. If any nodes that run an older IBM Spectrum Scale version have mounted the file system at the time this command is entered, the `mmchfs` command fails. This consideration might also apply to minor releases within the same major release. For example, release 5.0.2.x and release 5.1.0.x might have different metadata formats.

To enable backward-compatible format changes, enter the following command:

```
mmchfs FileSystem -V compat
```

To upgrade the desired file system to the latest metadata format changes, enter the following command:

```
mmchfs FileSystem -V full
```

Certain new file system features might require more processing that cannot be handled by the `mmchfs -V` command alone. To fully activate such features, in addition to `mmchfs -V`, you must also run the `mmigratefs` command.

Note: The first mount of a file system after you run `mmchfs -V` might fail with a no-disk-space error. This situation might occur if the file system is relatively low on metadata disk space (10% or less free). If so, enter the command again. Typically the file system is mounted without a problem after the initial failed mount.

After completing the upgrade, you might need to enable cluster configuration repository (CCR) and fast extended attributes (fastea).

5. Enable cluster configuration repository (CCR) and fast extended attributes (fastea) as follows, if required.

- a) Enable the CCR in the cluster as follows.

```
mmchcluster --ccr-enable
```

- b) Enable fastea in the cluster as follows.

i) Check if fastea is enabled.

```
mmfsfs FileSystemName --fastea
```

A sample output is as follows.

```
flag value description -----
attributes enabled?          --fastea Yes Fast external
```

ii) If fastea is not enabled, issue the following command to enable it.

```
mmmigratefs FileSystemName --fastea
```

For more information, see *mmmigratefs command* in *IBM Spectrum Scale: Command and Programming Reference*.

Note: The following features are a few among several IBM Spectrum Scale features that require fast extended attributes (fastea) to be enabled to work.

- Clones
- Independent filesets
- Encryption
- Active file management (AFM)

6. If you have file audit logging or clustered watch folder enabled on any file systems, follow the steps in [“Upgrade paths and commands for file audit logging and clustered watch folder”](#) on page 542.

Note: Ensure that you include the `gpfs.librdkafka` package as part of the upgrade process if those packages are currently installed.

7. Set the value of the `tscCmdAllowRemoteConnections` attribute to no.

The `tscCmdAllowRemoteConnections` attribute specifies whether the `ts*` commands are allowed to use the remote TCP/IP connections when communicating with the local or other `mmfsd` daemons. The scope of the `tscCmdAllowRemoteConnections` attribute is local to a cluster and it has the same value on all the nodes in the cluster.



Warning: Make sure that all remote clusters must be at 5.1.3 or later. Setting the `tscCmdAllowRemoteConnections` parameter to no impacts the remote cluster functions, hence the recommendation is to check the version of the remote clusters. If any remote cluster is running a version of IBM Spectrum Scale older than 5.1.3, setting the `tscCmdAllowRemoteConnections` parameter value to no in the home cluster can generate `Operation not permitted` errors. For more information, see *mmchconfig command* in *IBM Spectrum Scale: Command and Programming Reference*.

Reverting to the previous level of IBM Spectrum Scale

If you decide not to continue the upgrade to the latest level of GPFS, and you have not yet issued the `mmchfs -V` command, you can reinstall the earlier level of GPFS.

Important: Once a file system has been upgraded explicitly by issuing the `mmchfs -V full` command, the disk images can no longer be read by a prior version of GPFS. You must re-create the file system from the backup media and restore the content if you choose to go back after this command is issued. The same rules apply for file systems that are newly created with GPFS 5.0.

You can revert back to GPFS 4.2.x.

If you have performed backups with the `mmbackup` command using the 5.0 version and decide to revert to an earlier version, you must rebuild the `mmbackup` shadow database using the `mmbackup` command with either the `-q` or `--rebuild` option.

The procedure differs depending on whether you have issued the `mmchconfig release=LATEST` command or not.

Reverting to a previous level of GPFS when you have *not* issued mmchconfig release=LATEST

If you have not issued the `mmchconfig release=LATEST` command, perform these steps.

1. Stop all user activity in the file systems.
2. Cleanly unmount all mounted GPFS file systems. Do not use force unmount.
3. Stop GPFS on all nodes in the cluster:

```
mmshutdown -a
```

4. Run the appropriate uninstallation program to remove GPFS from each node in the cluster. For example:

- For Linux nodes (this example is only applicable on the IBM Spectrum Scale Standard Edition):

```
rpm -e gpfs.gpl gpfs.license.std gpfs.base gpfs.docs gpfs.gskit gpfs.msg.en_US
```

- For AIX nodes:

```
installp -u gpfs
```

- For Windows nodes, open the Programs and Features control panel and remove IBM General Parallel File System.

For the remaining steps, see [IBM Spectrum Scale in IBM Documentation](#), and search for the appropriate *IBM Spectrum Scale: Concepts, Planning, and Installation Guide* for your release.

5. Copy the installation images of the previous GPFS licensed program on all affected nodes.
6. Install the original install images and all required PTFs.
7. For Linux nodes running GPFS, you must rebuild the GPFS portability layer.
8. Reboot all nodes.

Related tasks

[Reverting to a previous level of GPFS when you have issued mmchconfig release=LATEST](#)

If you *have* issued the `mmchconfig release=LATEST` command, you must rebuild the cluster. Perform these steps.

Reverting to a previous level of GPFS when you *have* issued mmchconfig release=LATEST

If you *have* issued the `mmchconfig release=LATEST` command, you must rebuild the cluster. Perform these steps.

1. Stop all user activity in the file systems.
2. Cleanly unmount all mounted GPFS file systems. Do not use force unmount.
3. Stop GPFS on all nodes in the cluster:

```
mmshutdown -a
```

4. Export the GPFS file systems by issuing the `mmexportfs` command:

```
mmexportfs all -o exportDataFile
```

5. Delete the cluster:

```
mmdeinode -a
```

6. Run the appropriate uninstallation command to remove GPFS from each node in the cluster. For example:

- For Linux nodes:

```
rpm -e gpfs.gpl gpfs.license.xx gpfs.base gpfs.docs gpfs.msg.en_US
```

- For AIX nodes:

```
installp -u gpfs
```

- For Windows nodes, open the Programs and Features control panel and remove IBM General Parallel File System.

For the remaining steps, see [IBM Spectrum Scale in IBM Documentation](#), and search for the appropriate *IBM Spectrum Scale: Concepts, Planning, and Installation Guide* for your release.

7. Copy the installation images of the previous GPFS licensed program on all affected nodes.
8. Install the original installation images and all required PTFs.
9. For Linux nodes running GPFS, you must rebuild the GPFS portability layer.
10. Reboot all nodes.
11. Recreate your original cluster using the `mmcrcluster` command. Run the `mmchlicense` command to set the appropriate licenses after the cluster is created.
12. Use the `mmchconfig` command to restore any previously set configuration settings that are compatible with GPFS 4.2 or below.
13. Import the file system information using the `mmimportfs` command. Specify the file created by the `mmexportfs` command from Step “4” on page 578:

```
mmimportfs all -i exportDataFile
```

14. Start GPFS on all nodes in the cluster, issue:

```
mmstartup -a
```

15. Mount the file systems if this is not done automatically when the GPFS daemon starts.

Related tasks

Reverting to a previous level of GPFS when you have not issued `mmchconfig release=LATEST`
If you have not issued the `mmchconfig release=LATEST` command, perform these steps.

Coexistence considerations

Each GPFS cluster can have multiple GPFS file systems that coexist on the cluster, but function independently of each other. In addition, each file system might have different data management programs.

Note: The GPFS Data Management API (DMAPI) and GPFS file system snapshots can coexist; however, access to the files in a snapshot using DMAPI is restricted. For more information, see *IBM Spectrum Scale: Command and Programming Reference*.

Compatibility considerations

All applications that ran on the previous release of GPFS will run on the new level of GPFS. File systems that were created under the previous release of GPFS can be used under the new level of GPFS.

Considerations for IBM Spectrum Protect for Space Management

Upgrading to GPFS 3.3 or beyond requires consideration for IBM Spectrum Protect for Space Management. IBM Spectrum Protect for Space Management requires that all nodes in a cluster are configured with a DMAPI file handle size of 32 bytes.

During upgrade, it is possible to have older versions of GPFS that have a DMAPI file handle size of 16 bytes. Until all nodes in the cluster have been updated to the latest release and the DMAPI file handle

size has been changed to 32 bytes, IBM Spectrum Protect for Space Management must be disabled. Run `mmlsconfig dmapiFileHandleSize` to see what value is set.

After all nodes in the cluster are upgraded to the latest release and you change the DMAPI file handle size to 32 bytes, you can enable IBM Spectrum Protect for Space Management.

The DMAPI file handle size is configured with the **dmapiFileHandleSize** option. For more information about this option, see *GPFS configuration attributes for DMAPI* in *IBM Spectrum Scale: Command and Programming Reference*.

Applying maintenance to your IBM Spectrum Scale system

Before applying maintenance to your IBM Spectrum Scale system, there are several things you should consider.

Remember that:

1. There is limited interoperability between IBM Spectrum Scale 5.1.x nodes and nodes running IBM Spectrum Scale 5.0.y. This function is intended for short-term use. You do not get the full functions of IBM Spectrum Scale 5.1.x until all nodes are using IBM Spectrum Scale 5.1.x.
2. Interoperability between maintenance levels is specified on a per-maintenance-level basis. See the [IBM Spectrum Scale FAQ](#) in IBM Documentation.
3. When applying maintenance, on the node being upgraded, the GPFS file systems need to be unmounted and GPFS must be shut down. Any applications using a GPFS file system must be shut down before applying maintenance to avoid application errors or possible system halts.

Note: If the AIX Alternate Disk Install process is used, the new image is placed on a different disk. The running install image is not changed, and GPFS is not shut down. The new image is not used until the node is rebooted.

4. For the latest service information, see the [IBM Spectrum Scale FAQ](#) in IBM Documentation.

To download fixes, go to the IBM Support Portal: Downloads for General Parallel File System (www.ibm.com/support/entry/portal/Downloads/Software/Cluster_software/General_Parallel_File_System) and obtain the fixes for your hardware and operating system.

To apply the latest fixes:

- For Linux nodes, see Chapter 4, “Installing IBM Spectrum Scale on Linux nodes and deploying protocols,” on page 351.
- For AIX nodes, see Chapter 5, “Installing IBM Spectrum Scale on AIX nodes,” on page 453.
- For Windows nodes, see Chapter 6, “Installing IBM Spectrum Scale on Windows nodes,” on page 457.

Guidance for upgrading the operating system on IBM Spectrum Scale nodes

This is a high-level guidance for upgrading the operating system (OS) on your IBM Spectrum Scale nodes with focus on steps to be performed from the IBM Spectrum Scale side.

For OS specific upgrade steps, refer to the respective OS documentation.



Warning: If upgrading from an older version of OS to a newer version requires a re-installation of the OS, the devices that are used for the install disk must be reviewed carefully before accepting the disk presented by the installer. This is required to avoid situations in which the installer chooses the IBM Spectrum Scale NSDs and causes file system corruption. You must ensure that IBM Spectrum Scale NSDs are not altered during OS re-installation or upgrade.

Order of upgrade

Important: Before upgrading the OS on a node, ensure that the new OS version that you are planning to upgrade to is supported on the current IBM Spectrum Scale version. For information on the supported OS versions, see [Functional Support Matrices](#) in *IBM Spectrum Scale FAQ*.

- If the new OS version that you are planning to upgrade to is supported on the current IBM Spectrum Scale version, upgrading IBM Spectrum Scale is not mandatory. Proceed with upgrading the OS.
- If the new OS version that you are planning to upgrade to is not supported on the current IBM Spectrum Scale version, do the following:
 1. Determine the IBM Spectrum Scale version on which the new OS version that you are planning to upgrade to is supported by referring to [Functional Support Matrices in IBM Spectrum Scale FAQ](#).
 2. Upgrade IBM Spectrum Scale.
 3. After the IBM Spectrum Scale upgrade is completed, upgrade to the new OS version.

Prerequisites for upgrading Linux distributions

- Download and mount the operating system installation image on each node that is being upgraded.
- Create an OS base repository that points to the installation image on each node that is being upgraded.
 - If the repository name is the same as an old OS repository, clean the repository cache.
 - Verify that the repository works.
- If required, download OFED drivers on each node that is being upgraded.
- Note the current OS and OFED settings. You might need to restore them after the OS upgrade.

It is assumed that mixed level of OS and drivers across nodes are supported by vendors.

- Note the current SELinux and firewall settings. Although an OS upgrade typically does not change these settings, it is recommended to note these settings.
- Check the file system space and ensure that /boot has adequate free space.
- If you are upgrading protocol nodes to Red Hat Enterprise Linux 8.x, you must perform a cleanup of object services and the related packages before attempting the upgrade. For more information, see [“Instructions for removing object protocol packages when upgrading protocol nodes to Red Hat Enterprise Linux 8.x” on page 585](#).

OS upgrade flow

Do the following steps on each node on which the OS is being upgraded.

1. If a protocol node is being upgraded, do the following:
 - a. Suspend the protocol node and stop the protocol services running on the node by issuing the following command.

```
mmces node suspend --stop
```

- b. Wait for the protocol service monitor to change to the suspended state.

# mmhealth node show ces	Component	Status	Status Change	Reasons
CES	SUSPENDED	30 min. ago	-	
AUTH	SUSPENDED	30 min. ago	-	
AUTH_OBJ	SUSPENDED	30 min. ago	-	
BLOCK	SUSPENDED	30 min. ago	-	
CESNETWORK	SUSPENDED	30 min. ago	-	
NFS	SUSPENDED	30 min. ago	-	
OBJECT	SUSPENDED	31 min. ago	-	
SMB	SUSPENDED	30 min. ago	-	

2. Disable the GPFS autoload setting that causes GPFS to be started whenever the operating system reboots by issuing the following command.

```
mmchconfig autoload=no -N NodeName
```

3. Unmount GPFS file systems on the node by issuing the following command.

```
mmumount all
```

4. Stop GPFS on the node by issuing the following command.

```
mmshutdown
```

5. Check if there are any residual GPFS processes running and kill them, if needed.
6. If required, uninstall the OFED driver using the bundled uninstallation script in the OFED package.
7. Perform the OS upgrade steps according to the respective documentation.
8. Reboot the node when the OS upgrade steps are completed.
9. If required, install the new OFED driver and reboot the node after OFED driver installation.
10. Restore any OS and OFED settings (if applicable) that have been overwritten during the upgrade.
11. If a `leapp` upgrade from Red Hat Enterprise Linux 7.x to 8.x is performed, make sure to restore some of the removed packages and follow the details in [“Guidance for Red Hat Enterprise Linux 8.x on IBM Spectrum Scale nodes”](#) on page 583.
12. Rebuild the GPFS portability layer. For more information, see [“Building the GPFS portability layer on Linux nodes”](#) on page 364.
13. Verify that no conflicting services have been started or enabled during the OS upgrade. For example, some versions of RHEL autostart a non-IBM version of NFS after upgrade and it conflicts with the CES NFS service of IBM Spectrum Scale.
14. Verify SELinux and firewall settings.
15. Enable the GPFS autoloading setting, if it was disabled earlier in this procedure, by issuing the following command.

```
mmchconfig autoloading=yes -N NodeName
```

16. Restart GPFS on the node by issuing the following command.

```
mmstartup
```

17. Verify that all file systems are mounted on the node by issuing the following command.

```
mmismount all
```

18. If a protocol node is being upgraded, do the following:

- a. Resume the protocol node and start the protocol services on the node by issuing the following command.

```
mmces node resume --start
```

- b. Ensure that CES IPs are hosted by this node and that all services are healthy by issuing the following commands.

```
mmces address list
mmces service list -a
mmces node list
mmhealth node show CES
mmhealth cluster show
```

Repeat these steps on all nodes on which the OS needs to be upgraded.

Related tasks

[“Upgrading the SMB package after upgrading OS”](#) on page 523

If you have upgraded the operating system on your protocol nodes, ensure that the SMB package is upgraded to the version required for the new OS.

[“Upgrading the performance monitoring packages after upgrading Ubuntu OS”](#) on page 527

If you have upgraded the Ubuntu operating system on your IBM Spectrum Scale nodes, additional steps are required to ensure that performance monitoring packages are upgraded to the version required for the latest version of Ubuntu.

[“Servicing IBM Spectrum Scale protocol nodes” on page 589](#)

Use these steps to upgrade OS and to install kernel updates, firmware updates, and driver updates on IBM Spectrum Scale protocol nodes.

Guidance for Red Hat Enterprise Linux 8.x on IBM Spectrum Scale nodes

Red Hat Enterprise Linux 8.x is supported starting with IBM Spectrum Scale release 5.0.4.

Remember: Red Hat Enterprise Linux 8.0 is not an Extended Update Support (EUS) release, so the errata stream stops when Red Hat Enterprise Linux 8.1 is released. Do not use Red Hat Enterprise Linux 8.0 in production. Upgrade Red Hat Enterprise Linux to 8.1 to stay within the IBM Spectrum Scale support matrix.

The following list contains some general guidance when installing Red Hat Enterprise Linux 8.x on IBM Spectrum Scale nodes.

- After Red Hat Enterprise Linux 8.x is installed, IBM Spectrum Scale on x86 requires `elfutils-devel` to be installed for successfully building the GPFS portability layer (GPL) by using the `mmbuildgp1` command.
- Review the Red Hat Enterprise Linux 8.x documentation topic *Considerations for adopting RHEL8* for all hardware and driver support that was previously supported but has since been removed.
- Review the [IBM Spectrum Scale FAQ in IBM Documentation](#) for more information and for the supported kernel versions.

Red Hat provides a `leapp upgrade` utility to upgrade the OS from Red Hat Enterprise Linux 7.x to 8.x but it has some limitations. It is recommended to provision a new Red Hat Enterprise Linux 8.x node instance instead of using a `leapp upgrade` if possible. If you are using the `leapp upgrade` utility, there are some details to be aware of that are as follows.

- A `leapp upgrade` might upgrade the IBM Spectrum Scale node to the latest release of Red Hat Enterprise Linux 8.x. Ensure that `leapp upgrade` does not upgrade the node to an unsupported OS version. For supported OS and kernel versions, see [IBM Spectrum Scale FAQ in IBM Documentation](#).

Note: On s390x, a `leapp upgrade` from Red Hat Enterprise 7.x to 8.x is not supported. Using the `leapp` utility for the upgrade can lead to an unbootable system as described in the Red Hat bugzillas RH1890215 and RH1881428. Contact IBM support if you plan to upgrade from Red Hat Enterprise 7.x to 8.x on s390x.

- Using the `leapp upgrade` utility to upgrade from Red Hat Enterprise Linux 7.6 to 8.x with IBM Spectrum Scale 5.0.4.x or later installed might result in the removal of some IBM Spectrum Scale packages. This package removal might potentially happen with any third-party software. It is highly recommended to use `leapp-0.8.1-1` or later. The `leapp` utility is supported by Red Hat and any issues that are encountered with the `leapp upgrade` utility must be reported to Red Hat. Before you run `leapp upgrade`, save the list of IBM Spectrum Scale packages that are installed on the system by using the `rpm -qa | grep gpfs` command and then compare it to the packages installed after the upgrade. The `/var/log/leapp` directory also contains the packages that are to be removed.
- If Cluster Export Services (CES) are enabled on an IBM Spectrum Scale node running on a version earlier than 5.1.0 and you run a `leapp upgrade` from Red Hat Enterprise Linux 7.x to 8.x, then you must first stop object and remove its associated packages. For more information, see [“Instructions for removing object protocol packages when upgrading protocol nodes to Red Hat Enterprise Linux 8.x” on page 585](#).

Note: With certain versions of IBM Spectrum Scale, the object protocol is not supported on Red Hat Enterprise Linux 8.x. Refer to IBM Spectrum Scale FAQ in IBM Documentation for determining the IBM Spectrum Scale version on which the object protocol is supported on Red Hat Enterprise Linux 8.x and if you can reinstall object or not.

- If Cluster Export Services (CES) are enabled on an IBM Spectrum Scale 5.0.4.x or later node and you run a `leapp` upgrade from Red Hat Enterprise Linux 7.x to 8.x, the NFS el7 packages are removed. The NFS el8 packages (`gpfs.nfs-ganesha-*.rpm`, `gpfs.nfs-ganesha-gpfs-*.rpm`, `gpfs.nfs-ganesha-utils-*.rpm`) need to be installed after the `leapp` upgrade. The NFS el8 packages are located in the `/usr/lpp/mmfs/5.x.x.x/ganesha_rpms/rhel8` directory. You can use `yum localinstall` or `rpm -ivh` commands to install these packages.
- If Cluster Export Services (CES) are enabled on an IBM Spectrum Scale 5.0.4.x or later node and you run a `leapp` upgrade from Red Hat Enterprise Linux 7.x to 8.x, the SMB el7 package is removed. The SMB el8 package needs to be installed after the `leapp` upgrade. The SMB el8 package is located in the `/usr/lpp/mmfs/5.x.x.x/smb_rpms/rhel8` directory. You can use `yum localinstall` or `rpm -ivh` commands to install this package.
- If performance collection or GUI functionalities are enabled on an IBM Spectrum Scale 5.0.4.x or later node and you run a `leapp` upgrade from Red Hat Enterprise Linux 7.x to 8.x, the associated `gpfs.gss.pmcollector` or `gpfs.gui` packages are removed and they need to be reinstalled after the `leapp` upgrade. For more information, see [“Reinstalling performance monitoring collector on Red Hat Enterprise Linux 8.x nodes” on page 526](#).
 - Performance collection package can be installed from: `/usr/lpp/mmfs/5.x.x.x/zimon_rpms/rhel8/gpfs.gss.pmcollector`
 - If NFS is also installed, the NFS Ganesha performance monitoring package can be installed from: `/usr/lpp/mmfs/5.x.x.x/zimon_rpms/rhel8/gpfs.pm-ganesha`
 - Management GUI package can be installed from: `/usr/lpp/mmfs/5.x.x.x/gpfs_rpms/gpfs.gui*`

After installing the management GUI package, start the management GUI by using the **`systemctl start gpfsgui`** command.
- A `leapp` upgrade from Red Hat Enterprise Linux 7.x to 8.x removes the `python-ldap` package, which is prerequisite for the **`mmadquery`** command. Ensure that you install `python3-ldap` after `leapp` upgrade, if you plan to use **`mmadquery`** on Red Hat Enterprise Linux 8.x.
- The CLI utility **`mmadquery`** that is bundled with IBM Spectrum Scale 5.1.0 for Red Hat Enterprise Linux 7.x is based on python2. The CLI utility **`mmadquery`** that is bundled with IBM Spectrum Scale 5.1.0 for Red Hat Enterprise Linux 8.x is based on python3. When you install IBM Spectrum Scale 5.1.0 on Red Hat Enterprise Linux 7.x and upgrade an operating system from Red Hat Enterprise Linux 7.x to Red Hat Enterprise Linux 8.x, **`mmadquery`** still points to python2.

To point to the correct **`mmadquery`**, which is based on python3, perform the following steps:

1. Remove existing `/usr/lpp/mmfs/bin/mmadquery`.
 2. Copy `/usr/lpp/mmfs/bin/mmadquery_py3` to `/usr/lpp/mmfs/bin/mmadquery`.
- After all packages that were removed by `leapp` upgrade are reinstalled as described in the preceding text, complete the steps in [“Guidance for upgrading the operating system on IBM Spectrum Scale nodes” on page 580](#) to finish bringing the node online after the OS is upgraded to Red Hat Enterprise Linux 8.x.
 - If Cluster Export Services (CES) are enabled on an IBM Spectrum Scale node, an in-place upgrade from Red Hat Enterprise Linux 8.x by using `leapp` upgrade utility might fail.

The `leapp` utility does not support upgrading Red Hat Enterprise Linux 8.x on the file protocols nodes because of the winbind strings in the `/etc/nsswitch.conf` file. When either SMB or NFS or both SMB and NFS protocols are configured for Active Directory (AD) based authentication schemes by using the **`mmuserauth`** command, this command adds winbind in the `/etc/nsswitch.conf` file of each protocol node. The in-place upgrade with winbind strings in the `/etc/nsswitch.conf` is not allowed because of a known OS upgrade limitation.

To do the in-place upgrade, you need to complete the steps from 1 through 4 in the [Troubleshooting known issues](#) documentation for upgrading Red Hat Enterprise Linux 8.x. After the OS upgrade, when you resume the protocol nodes, the IBM Spectrum Scale adds the removed winbind entries in

the `/etc/nsswitch.conf` file on each protocol node. You can verify the successful addition of the removed winbind entries for one of the CES nodes by issuing the following command:

```
# mmuserauth service check -N cesNodes
```

After the OS upgrade, do the following steps to confirm correctness of the `/etc/nsswitch.conf` file on the nodes:

1. Check whether the current configuration is valid by issuing the following command:

```
# authselect check
```

2. Check whether the `/etc/nsswitch.conf` file is linked to the current authselect profile `nsswitch.conf` file by issuing the following command:

```
# ls -l /etc/nsswitch.conf
```

3. If either the current configuration is incorrect or the `/etc/nsswitch.conf` file is not linked to the current authselect profile, apply forcefully the current authselect profile on the nodes.

- a. Check the current authselect profile by issuing the following command:

```
# authselect current
```

- b. Apply forcefully the authselect profile again by issuing the following command:

```
# authselect select ProfileNameIdentifiedFromPreviousCommand --force
```

The packages that `leapp` upgrade might remove can vary depending on packages that are installed and dependencies that might exist. It might be required to reinstall any removed packages, which can typically be determined by viewing the log `/var/log/leapp/leapp-upgrade.log` and looking for *Removing dependent packages* after the `leapp` upgrade completes and the node restarts.

Instructions for removing object protocol packages when upgrading protocol nodes to Red Hat Enterprise Linux 8.x

The presence of the object packages and their dependencies can prevent upgrade to Red Hat Enterprise Linux® 8.x. Therefore, it is necessary to remove the associated packages from the system when you are upgrading protocol nodes to Red Hat Enterprise Linux® 8.x. After the upgrade is complete, the object packages can be reinstalled.

The object protocol is only supported on Red Hat Enterprise Linux® 8.x with IBM Spectrum Scale version 5.1.0.1 and later. Systems at earlier versions of IBM Spectrum Scale with object protocol need to upgrade to version 5.1.0.1 or later after the system upgrade to Red Hat Enterprise Linux® 8.x.



Attention: The object protocol is not supported in IBM Spectrum Scale 5.1.0.0. If you want to deploy object, upgrade to the IBM Spectrum Scale 5.1.0.1 or a later release.

1. If the object protocol is running in the cluster, stop it according to the following instructions.

For more information, see *Understanding and managing Object services in IBM Spectrum Scale: Administration Guide*.

2. After the object protocol is stopped, certain packages need to be cleaned up to prevent conflicts during the upgrade to Red Hat Enterprise Linux 8.x.

Run the following command to uninstall these packages and their dependencies:

```
yum erase isa-l liberasurecode openstack-keystone openstack-selinux openstack-swift-account  
openstack-swift-container openstack-swift-object  
openstack-swift-proxy python2-bcrypt python2-cinderclient python2-dogpile-cache python2-  
glanceclient python2-keystoneauth1 python2-keystoneclient python2-keystonemiddleware  
python2-neutronclient python2-novaclient python2-openstackclient python2-osc-lib python2-os-  
client-config python2-oslo-cache python2-oslo-concurrency python2-oslo-config  
python2-oslo-context python2-oslo-db python2-oslo-i18n python2-oslo-middleware python2-oslo-  
service python2-pbr python2-pysocks python2-repoze-lru python2-requests  
python2-requestsexceptions python2-scrypt python2-argparse python-cliff-tablib  
python-debtcollector python-dns python-funcsigs python-greenlet python-httplib2
```

```
python-jinja2 python-keystone python-msgpack python-openstackclient-lang python-oslo-cache-  
lang python-oslo-concurrency-lang python-oslo-db-lang python-oslo-i18n-lang  
python-oslo-middleware-lang python-pip python-posix_ipc python-pyeclib python-simplejson  
python-sqlalchemy python-swift python-tablib python-testtools python-wrapt  
python-wsgiref python-zope-event swift3 python-keyring python-idna python-PyMySQL
```

3. After the packages that are mentioned in the preceding step are removed, the following packages must be downgraded to the level of the system Red Hat Enterprise Linux packages by running the following command:

```
yum downgrade pyOpenSSL python2-cryptography
```

If these packages are not removed or downgraded, the upgrade to Red Hat Enterprise Linux 8.x using the **leapp** utility might fail with a message such as the following message:

```
[ERROR] Actor: prepare_upgrade_transaction Message: A Leapp Command Error occurred
```

The **leapp** command with the **--verbose** option might display messages such as the following message:

```
file *** from install of package ### conflicts with file from package ###
```

If the messages are related to the packages associated with the object protocol, performing the cleanup steps eliminates these errors.

After the object protocol is stopped and the associated packages are removed, proceed with the Red Hat Enterprise Linux 8.x upgrade.

Note: Due to the object packages being removed for the leapp upgrade, these special steps need to be done when you are using the installation toolkit to upgrade the object protocol to version 5.1.x after the system upgrade is complete:

1. Use the installation toolkit offline upgrade for the protocol nodes.
2. After the installation toolkit has successfully completed the upgrade, manually migrate the object configuration data to version 5.1 by using the following command:

```
mmobj config manage --version-sync
```

For more information, see [“Upgrading object packages to IBM Spectrum Scale 5.1.x or later by using the installation toolkit” on page 561.](#)

Considerations for upgrading from an operating system not supported in IBM Spectrum Scale 5.1.x.x

In IBM Spectrum Scale 5.1.0.x, there are several changes in the operating system support matrix. Therefore, note the considerations that might apply to your environment and then plan the upgrade.

Note: The IBM Spectrum Scale upgrade procedures and the guidance for the operating system upgrades remain largely unchanged and they can be used as before.

Use the installation toolkit for upgrading to IBM Spectrum Scale 5.1.x.x. You can also use the manual upgrade procedure, but it requires more steps to be done that the installation toolkit does automatically.

If you are upgrading from a release earlier than IBM Spectrum Scale 5.0.0, you can either use the offline upgrade procedure that requires complete cluster shutdown or do the upgrade in two hops.

For more information, see the following topics:

- [Upgrading overview](#)
- [“IBM Spectrum Scale supported upgrade paths” on page 506](#)
- [“Upgrading IBM Spectrum Scale components with the installation toolkit” on page 543](#)
- [Offline upgrade with complete cluster shutdown](#)

- [“Guidance for upgrading the operating system on IBM Spectrum Scale nodes” on page 580](#)
- [Which Linux distributions are supported by IBM Spectrum Scale? \[IBM Spectrum Scale FAQ\]](#)

The following considerations apply if you are upgrading from an operating system that is not supported in IBM Spectrum Scale 5.1.x.x:

- After you upgrade to the new operating system, you might need to reinstall packages specific to the new operating system for some components such as performance monitoring, NFS, SMB, or GUI.
- The object protocol is not supported on Ubuntu and Red Hat Enterprise Linux 7.x on IBM Spectrum Scale 5.1.x.x. If you want to continue to use the object protocol on Ubuntu or Red Hat Enterprise Linux 7.x operating systems, it is advised to continue on IBM Spectrum Scale 5.0.x. System upgrade paths are as follows:

Ubuntu 16.0.4 or 18.04

1. Disable object.
2. Upgrade to Ubuntu 20.04.
3. Upgrade to IBM Spectrum Scale 5.1.x.x.

Note: Object services will not be available after the upgrade.

Red Hat Enterprise Linux 7.6 or earlier 7.x

1. Stop object services.
 2. Clean up object packages. For more information, see [“Instructions for removing object protocol packages when upgrading protocol nodes to Red Hat Enterprise Linux 8.x” on page 585.](#)
 3. Upgrade protocol nodes to Red Hat Enterprise Linux 8.1 or later by using the Red Hat Leapp utility.
 4. Upgrade to IBM Spectrum Scale 5.1.x.x.
- If you have NFS or SMB enabled on SLES 12 nodes, you must disable NFS and SMB before you upgrade the operating system to SLES 15 and then you upgrade to IBM Spectrum Scale 5.1.x.x. You can enable SMB and NFS after these tasks are completed.
 - If you have NFS or SMB enabled on Ubuntu 16.04 or 18.04 nodes, you must disable NFS and SMB before you upgrade the operating system to Ubuntu 20.04 and then you upgrade to IBM Spectrum Scale 5.1.x.x. You can enable SMB and NFS after these tasks are completed.

Table 47. Upgrade paths if upgrade from OS not supported in IBM Spectrum Scale 5.1.x.x

Current IBM Spectrum Scale version	Current operating system version to target operating system version	Recommended course of action for upgrade
Version 5.0.0.0 or later	SLES 12 to SLES 15	<ol style="list-style-type: none"> 1. Upgrade the operating system. 2. Upgrade to IBM Spectrum Scale 5.1.x.x by using the installation toolkit.
	Ubuntu 16.04 or 18.04 to Ubuntu 20.04	<ol style="list-style-type: none"> 1. Upgrade the operating system. 2. Upgrade to IBM Spectrum Scale 5.1.x.x by using the installation toolkit.
	Red Hat Enterprise Linux 7.x (earlier than 7.6) to Red Hat Enterprise Linux 7.7 or later	<ol style="list-style-type: none"> 1. Upgrade the operating system. 2. Upgrade to IBM Spectrum Scale 5.1.x.x by using the installation toolkit.

Table 47. Upgrade paths if upgrade from OS not supported in IBM Spectrum Scale 5.1.x.x (continued)

Current IBM Spectrum Scale version	Current operating system version to target operating system version	Recommended course of action for upgrade
Version 4.2.3.x or earlier (4.2.0.x or later)	SLES 12 to SLES 15	<ol style="list-style-type: none"> 1. Upgrade to IBM Spectrum Scale 5.0.x.x by using the installation toolkit. 2. Upgrade the operating system. 3. Upgrade to IBM Spectrum Scale 5.1.x.x by using the installation toolkit. <p>You can also use the offline upgrade procedure to directly upgrade to 5.1.x.x after the operating system upgrade is done.</p>
	Ubuntu 16.04 to Ubuntu 20.04	<p>This upgrade path requires offline upgrade.</p> <ol style="list-style-type: none"> 1. Bring the cluster offline. 2. Upgrade the operating system. 3. Upgrade to IBM Spectrum Scale 5.1.x.x by using the installation toolkit. 4. Bring the cluster online.
	Red Hat Enterprise Linux 6.x or 7.x (version 7.6 or earlier) to Red Hat Enterprise Linux 7.7 or later	<ol style="list-style-type: none"> 1. Upgrade to IBM Spectrum Scale 5.0.x.x by using the installation toolkit. 2. Upgrade the operating system. 3. Upgrade to IBM Spectrum Scale 5.1.x.x by using the installation toolkit. <p>You can also use the offline upgrade procedure to directly upgrade to 5.1.x.x after you upgrade the operating system.</p>

Servicing IBM Spectrum Scale protocol nodes

Use these steps to upgrade OS and to install kernel updates, firmware updates, and driver updates on IBM Spectrum Scale protocol nodes.

It is advisable to match OS, kernel, firmware, and driver across all nodes within a cluster to help with performance and to ease debugging. As a part of this procedure, ensure the following:

- Before upgrading OS or installing kernel updates, refer to the IBM Spectrum Scale FAQ to ensure that the version of IBM Spectrum Scale currently running is supported on the version of OS that you are

planning to upgrade to or the kernel update that you are planning to install: [IBM Spectrum Scale FAQ in IBM Documentation](#).

It is possible that an IBM Spectrum Scale upgrade might be required to get to a version of IBM Spectrum Scale that supports the newly desired OS. If this is the case, upgrade IBM Spectrum Scale first and then the OS.

- If kernel update is for a new OS (RHEL 7.3 vs RHEL 7.2, for example), always update the OS before applying the kernel update.
- When taking nodes offline to upgrade OS or installing kernel updates, firmware updates, or driver updates, ensure the following:
 - Quorum does not break
 - Enough NSD nodes remain up to access NSDs
 - The remaining nodes can handle the desired workload

1. Uninstall the applicable drivers as follows.

- a) Obtain and extract the applicable drivers.
- b) Suspend CES on the node being upgraded.

```
mmces node suspend -N NodeBeingUpgraded --stop
```

c) Shut down GPFS on the node being upgraded.

```
mmshutdown -N NodeBeingUpgraded
```

d) Find the uninstallation script within the drivers package and execute it on the node being upgraded.

2. Create a local repository for the OS upgrade.

A repository must be created so that the OS can be upgraded. This repository can be DVD or ISO based. Make sure that you remove any repositories pointing to old OS versions.

3. Upgrade the OS and reboot the node after the OS upgrade.

For more information, see [“Guidance for upgrading the operating system on IBM Spectrum Scale nodes”](#) on page 580.

4. Install the kernel update and reboot the node after the kernel update.

5. Rebuild the GPFS portability layer using the **mmbuildgp1** command.

For more information, see [“Building the GPFS portability layer on Linux nodes”](#) on page 364.

6. Install the required firmware updates and reboot the node after the firmware update.

7. Install the latest drivers, as applicable and reboot the node after the driver update.

8. Verify that GPFS is active on the node and then resume CES.

```
mmgetstate -a
mmces node resume -N NodeBeingUpgraded --start
mmces node list
mmces service list -a
mmces address list
```

9. Repeat the preceding steps on each protocol node that needs to be serviced, one node at a time.

Offline upgrade with complete cluster shutdown

You can use the following steps for an offline upgrade.

Important: Online upgrade from an unsupported version of IBM Spectrum Scale or GPFS is not supported. However, you can use the following manual offline upgrade procedure to upgrade to a supported version of IBM Spectrum Scale.

1. Stop all file system operations on the cluster and any external access to the cluster. For example, protocol clients.

2. If you have protocol nodes then stop all CES operations.

```
mmces stop Protocol -a
```

3. Unmount all file systems.

```
mmumount all -a
```

4. Shut down GPFS on all nodes.

```
mmshutdown -a
```

Verify that the GPFS daemon has terminated and that the kernel extensions are unloaded by using **mmfsenv -u**. If the **mmfsenv -u** command reports that it cannot unload the kernel extensions because they are busy, then the installation can proceed, but the node must be rebooted after the installation. Kernel extensions are busy means that a process has a current directory in some GPFS file system directory or it has an open file descriptor. The Linux utility **lsof** can identify the process and that process can then be killed. Retry **mmfsenv -u** after killing the process and if the command succeeds, then a reboot of the node can be avoided.

Note:

- The **mmfsenv -u** command is normally used with assistance from support or for cases where its usage is specifically documented such as in this scenario.
- The **mmfsenv -u** command only checks the local node from which it is run, not all the nodes in the cluster.

Refer to the [IBM Spectrum Scale FAQ in IBM Documentation](#) for the supported operating system versions that apply to the IBM Spectrum Scale release that you are upgrading to.

5. If the operating system needs to be upgraded, then complete that upgrade.
6. Extract the IBM Spectrum Scale installation image so that all installation files (rpm, deb, bff, msi, and so on) are available.
7. Run the necessary platform-specific commands to upgrade GPFS.
 - SLES and RHEL Linux:

```
rpm -Fvh /usr/lpp/mmfs/5.x.y.z/gpfs_rpms/  
<OS>{gpfs.adv*.rpm,gpfs.base*.rpm,gpfs.crypto*.rpm,  
gpfs.docs*.rpm,gpfs.gpl*.rpm,gpfs.gskit*.rpm,gpfs.msg*.rpm,  
gpfs.license.XX*.rpm,gpfs.compression*.rpm}  
rpm -Fvh /usr/lpp/mmfs/5.x.y.z/zimon_rpms/  
<OS>{gpfs.gss.pmsensors*.rpm,gpfs.gss.pmcollector*.rpm}
```

Where <OS> is the operating system specific directory such as `rhel17` or `sles15`.

- The `gpfs-adv` and `gpfs-crypto` packages are available only with IBM Spectrum Scale Advanced Edition, IBM Spectrum Scale Data Management Edition, or IBM Spectrum Scale Developer Edition.
 - Depending on the IBM Spectrum Scale edition that you are installing, install the appropriate `gpfs.license` package.
 - IBM Spectrum Scale Developer Edition is available only on Red Hat Enterprise Linux on x86_64.
- Ubuntu:

```
dpkg -i /usr/lpp/mmfs/5.x.y.z/gpfs_debs/  
<OS>{gpfs.adv*.deb,gpfs.base*.deb,gpfs.crypto*.deb,  
gpfs.docs*.deb,gpfs.gpl*.deb,gpfs.gskit*.deb,gpfs.msg*.deb,  
gpfs.license.XX*.deb,gpfs.compression*.deb}  
  
dpkg -i /usr/lpp/mmfs/5.x.y.z/zimon_debs/  
<OS>{gpfs.gss.pmsensors*.deb,gpfs.gss.pmcollector*.deb}
```

Where <OS> is the operating system specific directory such as `ubuntu18`.

Note:

- The `gpfs-crypto` and the `gpfs-adv` packages are available only with IBM Spectrum Scale Advanced Edition and IBM Spectrum Scale Data Management Edition.
- Depending on the IBM Spectrum Scale edition that you are installing, install the appropriate `gpfs.license` package.

- AIX:

```
installp -agXYd . gpfs
```

- Windows:

- Uninstall the version of GPFS or IBM Spectrum Scale that you are upgrading from and reboot.
 - Uninstall the license of GPFS or IBM Spectrum Scale version that you are upgrading from .
 - Uninstall Open Secure Shell for GPFS on Windows, if applicable.
 - Disable any SUA daemon, such as OpenSSH, that might have been configured. Do *not* uninstall SUA yet, or you may lose GPFS configuration information.
 - Install the 64-bit version of Cygwin. For more information, see [“Installing Cygwin” on page 466](#).
 - Install `gpfs.base-5.1.x.x-Windows-license.msi`.
 - Install `gpfs.base-5.1.x.x-Windows.msi`.
 - Install IBM® GSKit for GPFS.
 - Uninstall SUA completely.
 - Configure passwordless SSH in Cygwin for mixed (Linux, AIX, Windows) clusters or use **mmwinserctl** on a Windows only cluster.
- On Linux systems, rebuild the GPFS portability layer (GPL) with the **mmbuildgpl** command. For more information, see [“Building the GPFS portability layer on Linux nodes” on page 364](#).
 - Restart GPFS by issuing the **mmstartup** command and remount file systems by issuing the **mmm mount service start Protocol -a** command.
- Note:** At this point, all code is at the upgraded level, but the cluster is not upgraded and new functions cannot be configured until the upgrade is completed.
- Complete the upgrade as documented in [“Completing the upgrade to a new level of IBM Spectrum Scale” on page 572](#).

Related concepts

[“IBM Spectrum Scale supported upgrade paths” on page 506](#)

Use this information to understand the supported upgrade paths for IBM Spectrum Scale.

[“Performing offline upgrade or excluding nodes from upgrade by using installation toolkit” on page 554](#)

Accessibility features for IBM Spectrum 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 Spectrum 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 Spectrum 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 \(www.ibm.com/software/globalization/terminology\)](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 Spectrum 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 Spectrum Scale: Command and Programming Reference*.

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 Spectrum 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 Spectrum Scale: Command and Programming Reference*.

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 Spectrum 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 Spectrum 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 filesset

A filesset 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 filesset, which enables more efficient per-filesset functions.

ISKLM

IBM Security Key Lifecycle Manager. For GPFS encryption, the ISKLM is used as an RKM server to store MEKs.

J

journaled 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 filesset to the root directory of another filesset.

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**

Disks 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 Spectrum 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 Spectrum 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 Spectrum Scale file systems. For more information, a sample program is provided in the following directory on IBM Spectrum Scale nodes: `/usr/lpp/mmfs/samples/util` called `tswf` that can be modified according to the user's needs.

Index

Special Characters

- ?
 - deleting old sensors [538](#)
- .afm
 - internal directory [53](#)
- .pconflicts
 - internal directory [53](#)
- .ptrash
 - internal directory [53](#)
- (excluding a node
 - offline upgrade [554](#)
- (marking a component offline(
 - during upgrade [554](#)
- /tmp/mmfs
 - collecting problem determination data in [349](#)

Numerics

- 2 K metadata [135](#)

A

- access control lists (ACLs)
 - file system authorization [262](#)
- access control on GPFS file systems
 - Windows [464](#)
- access to file systems
 - simultaneous [2](#)
- accessibility features for IBM Spectrum Scale [593](#)
- Active SMB connections
 - determining factors [310](#)
- adapter
 - invariant address requirement [219](#)
- adding Cloud servicesnode
 - existing Cloud servicescluster [478](#)
- adding LTFS nodes [446](#)
- adding protocol nodes
 - with toolkit [442](#)
- administration commands
 - GPFS [4](#), [17](#)
- AFM
 - AFM gateway nodes requirements on the cache and home clusters [342](#)
 - AFM to COS replication [129](#)
 - cache eviction [73](#)
 - caching modes [40](#)
 - encryption [89](#)
 - Fast creates [56](#)
 - fileset disabling [89](#)
 - gateway node [343](#)
 - global namespace [45](#)
 - independent-writer (IW) [40](#)
 - install [483](#)
 - Limitations [94](#), [95](#), [107](#)
 - local-update (LU) [40](#)
 - operation with disconnected home [77](#)

AFM (continued)

- Parallel data transfer using multiple remote mounts [61](#)
- Parallel data transfers [59](#)
- partial file caching [63](#)
- peer snapshot [70](#)
- planned maintenance
 - IW cache [87](#)
- planning [341](#)
- prefetch [63](#)
- primary gateway [43](#)
- psnap [70](#)
- read-only (RO) [40](#)
- resync on SW filesets [82](#)
- single-writer (SW) [40](#)
- UID and GID requirements on the cache and home clusters [341](#)
- upgrade [518](#)
- WAN latency [39](#)
- workerThreads on cache cluster [342](#)
- AFM directory [135](#)
- AFM DR
 - best practices [114](#)
 - cache cluster
 - UID/GID requirements [343](#)
 - changing
 - secondary site [105](#), [106](#)
 - characteristics
 - active-passive relationships [110](#)
 - independent fileset [109](#)
 - NFD exports [110](#)
 - NFSv3 [110](#)
 - one-on-one relationships [110](#)
 - failback
 - new primary site [105](#)
 - old primary site [104](#)
 - failover
 - secondary site [104](#)
 - features [99](#)
 - Immutability and appendOnly [99](#)
 - installing
 - AFM-based disaster recovery [485](#)
 - Limitations [95](#), [107](#)
 - planning [343](#)
 - role reversal [102](#)
 - RPO snapshots [100](#)
 - secondary cluster
 - NFS setup [344](#)
 - trucking
 - inband [111](#)
 - outband [111](#)
 - worker1Threads on primary cluster [343](#)
- AFM DR characteristics
 - failback [113](#)
 - failover [113](#)
 - trucking
 - inband [111](#)

- AFM DR characteristics (*continued*)
 - trucking (*continued*)
 - outband [111](#)
- AFM gateway node [11](#), [343](#)
- AFM gateway nodes requirements on the cache and home clusters [342](#)
- AFM mode
 - operations [48](#)
- AFM node [11](#)
- AFM to cloud object services replication
 - Limitations [129](#)
- AFM to cloud object storage [135](#), [139](#)
- AIX
 - electronic license agreement [454](#)
 - installation instructions for GPFS [453](#)
 - installing GPFS [453](#)
 - prerequisite software [453](#)
- allocation map
 - block [13](#)
 - inode [13](#)
 - logging of [13](#)
- allowing the GPFS administrative account to run as a service, Windows [471](#)
- antivirus software
 - Windows [462](#)
- API
 - configuring [479](#)
 - fields parameter [152](#)
 - filter parameter [154](#)
 - installing [479](#)
 - paging [157](#)
- application programs
 - communicating with GPFS [18](#)
- applying maintenance levels to GPFS [580](#)
- architecture, GPFS [8](#)
- assigning a static IP address
 - Windows [466](#)
- asynchronous
 - delay [48](#)
 - operations [48](#)
- asynchronous delay
 - AFM [48](#)
- atime value [260](#)
- audit logging
 - installation [422](#)
- authentication
 - basic concepts [291](#)
 - protocol user authentication [291](#)
- authentication planning
 - file access [294](#)
 - object access [299](#)
 - protocols [291](#), [293](#)
- autoload attribute [236](#)
- automatic mount
 - shared file system access [3](#)

B

- Backup considerations [340](#)
- backup planning [270](#)
- bandwidth
 - increasing aggregate [2](#)
- best practices [109](#)
- bin directory [354](#)

- block
 - allocation map [13](#)
 - size [258](#)
- block allocation map [261](#)

C

- cache [14](#)
- cache and home
 - AFM [41](#)
- cache cluster
 - AFM [39](#)
 - UID/GID requirements [343](#)
- cache eviction [73](#)
- cached
 - files [47](#)
- cached and uncached files
 - AFM [47](#)
- caching modes [40](#)
- call home
 - configuration [424](#)
 - event based uploads [192](#)
 - heartbeats [194](#)
 - installation [424](#)
 - mmcallhome [192](#), [194](#), [207](#)
 - monitoring IBM Spectrum Scale system remotely [192](#), [194](#), [207](#)
- Call home
 - Upgrade
 - v4.2.0. to v4.2.1 or later [525](#)
 - v4.2.1. to v5.0.0 [524](#)
- case sensitivity
 - Windows [461](#)
- CCR [26](#)
- CES
 - data uploads [140](#)
 - mmces command [140](#)
 - overview [29](#)
 - protocol [140](#)
 - SMB limitations
 - exports [306](#)
- CES HDFS support
 - planning [329](#)
- CES IPs [419](#)
- CES NFS limitations [304](#)
- CES NFS Linux limitations [304](#)
- CES NFS support
 - overview [31](#)
- CES shared root file system [417](#)
- changing
 - secondary site [105](#), [106](#)
- characteristics [109](#)
- clean up
 - Cloud services [503](#)
- clean up the environment [503](#)
- cleanup procedures for installation [496](#)
- cloud data sharing
 - creating a node class [475](#)
 - security considerations [338](#)
- Cloud data sharing
 - operation [171](#)
- cloud providers
 - Cloud services [175](#)

- cloud providers (*continued*)
 - transparent cloud tiering [175](#)
- Cloud services
 - creating a ndoe class [475](#)
 - modifying [478](#)
 - planning [329](#)
 - security considerations [338](#)
 - software requirements [329](#)
 - upgrade [531](#)
 - upgrade to 1.1.2.1 [532](#)
 - upgrade to 1.1.3 [532](#), [533](#)
 - upgrade to 1.1.4 [534](#), [535](#)
 - upgrade to 1.1.5 [536](#)
 - upgrade to 1.1.6 [537](#)
- cluster configuration data files
 - /var/mmfs/gen/mmldrfs file [25](#)
 - content [25](#)
- Cluster Export Services
 - overview [29](#)
- cluster manager
 - description [9](#)
 - initialization of GPFS [18](#)
- cluster node
 - considerations [331](#)
- Clustered configuration repository [26](#)
- clustered watch folder
 - events [185](#)
 - introduction [184](#)
 - JSON [186](#)
 - limitations [491](#)
 - remotely mounted file systems [492](#)
 - requirements [491](#)
- coexistence considerations [579](#)
- Coherency option [312](#)
- collecting details [188](#)
- collecting problem determination data [349](#)
- commands
 - description of GPFS commands [4](#)
 - failure of [232](#)
 - mmbackup [26](#)
 - mmcallhome [524–526](#)
 - mmchcluster [233](#), [234](#)
 - mmchconfig [15](#), [16](#)
 - mmchdisk [22](#)
 - mmcheckquota [18](#)
 - mmchfs [251](#), [262](#), [265](#), [304](#)
 - mmcrcluster [16](#), [228](#), [233](#), [234](#), [453](#)
 - mmcrfs [251](#), [262](#), [265](#), [304](#)
 - mmcrnsd [239](#), [240](#)
 - mmdefedquota [265](#), [266](#)
 - mmdefquotaon [266](#)
 - mmdelnsd [239](#), [240](#)
 - mmedquota [265](#)
 - mmfsck [13](#), [18](#), [22](#)
 - mmlsdisk [22](#), [248](#)
 - mmlsfs [13](#), [304](#)
 - mmlsquota [265](#)
 - mmmount [18](#)
 - mmrepquota [265](#)
 - mmstartup [236](#)
 - mmwinservctl [233](#), [234](#)
 - operating system [18](#)
 - processing [22](#)

- commands (*continued*)
 - remote file copy
 - rcp [234](#)
 - scp [234](#)
 - remote shell
 - rsh [233](#)
 - ssh [233](#)
- communication
 - cache and home [41](#)
 - GPFS daemon to daemon [230](#)
 - invariant address requirement [219](#)
- comparison
 - live system backups [275](#)
 - snapshot based backups [275](#)
- compatibility considerations [579](#)
- concepts
 - AFM DR [99](#)
- configuration
 - files [25](#)
 - flexibility in your GPFS cluster [3](#)
 - of a GPFS cluster [228](#)
- configuration and tuning settings
 - configuration file [237](#)
 - default values [237](#)
 - GPFS files [4](#)
- configuration of protocol nodes
 - object protocol [448](#), [449](#), [451](#)
- configuration tasks
 - SMB export limitations [306](#)
- configuring
 - api [479](#)
- configuring a mixed Windows and UNIX cluster [470](#)
- configuring a Windows HPC server [474](#)
- configuring GPFS [409](#)
- configuring quota
 - transparent cloud tiering [341](#)
- configuring system for transparent cloud tiering [332](#)
- configuring the GPFS Administration service, Windows [472](#)
- configuring Windows [466](#)
- considerations for backup
 - Transparent Cloud Tiering [340](#)
- considerations for GPFS applications
 - exceptions to Open Group technical standards [346](#)
 - NFS V4 ACL [346](#)
 - stat() system call [346](#)
- controlling the order in which file systems are mounted [268](#)
- conversion
 - mode [53](#)
- conversion of mode
 - AFM [53](#)
- created files (maximum number) [267](#)
- creating a node class
 - transparent cloud tiering [475](#)
- creating GPFS directory
 - /tmp/gpfs1pp on AIX nodes [454](#)
- creating the GPFS administrative account, Windows [471](#)
- creating the GPFS directory
 - /tmp/gpfs1pp on Linux nodes [355](#)

D

- daemon

- daemon (*continued*)
 - communication [230](#)
 - description of the GPFS daemon [5](#)
 - memory [14](#)
 - quorum requirement [9](#)
 - starting [236](#)
- DASD for NSDs, preparing [248](#)
- data
 - availability [2](#)
 - consistency of [2](#)
 - guarding against failure of a path to a disk [227](#)
 - maximum replicas [264](#)
 - recoverability [223](#)
 - replication [263](#), [264](#)
- data blocks
 - logging of [13](#)
 - recovery of [13](#)
- Data Management API (DMAPI)
 - enabling [266](#)
- data privacy
 - methodical command [208](#)
- data protection
 - backing up data [143](#)
 - commands [144](#)
 - create and maintain snapshots [144](#)
 - data mirroring [144](#)
 - fileset backup [143](#)
 - restoring a file system from a snapshot [144](#)
 - restoring data [144](#)
- data recovery
 - commands [144](#)
- data upload
 - scheduled upload [194](#)
- default data replication [264](#)
- default metadata replication [263](#)
- default quotas
 - description [266](#)
 - files [13](#)
- deleting authentication [301](#)
- deleting ID mapping [301](#)
- deleting old sensors
 - ? [538](#)
- deploying protocols
 - existing cluster [442](#)
- deploying protocols on Linux nodes
 - procedure for [416](#), [420](#), [421](#), [426](#), [543](#), [552](#)
- deployment considerations [109](#), [114](#)
- df command (specifying whether it will report numbers based on quotas for the fileset) [267](#)
- diagnosing errors [444](#)
- differences between GPFS and NTFS
 - Windows [463](#)
- direct access storage devices (DASD) for NSDs, preparing [248](#)
- directory, bin [354](#)
- Disable gpfs leases [311](#)
- disable leases [311](#)
- disabling protocols
 - authentication considerations [293](#)
- disabling the Windows firewall [466](#)
- disaster recovery
 - data mirroring [143](#)
 - protocol cluster disaster recovery [143](#)
 - SOBAR [143](#)
- disaster recovery (*continued*)
 - use of GPFS replication and failure groups [3](#)
- disk descriptor replica [248](#)
- disk usage
 - verifying [266](#)
- disks
 - considerations [238](#)
 - failure [226](#)
 - file system descriptor [11](#)
 - media failure [22](#)
 - mmcrfs command [257](#)
 - recovery [22](#)
 - releasing blocks [22](#)
 - stanza files [239](#)
 - state of [22](#)
 - storage area network [238](#)
 - stripe group [11](#)
- DMAPI
 - coexistence considerations [579](#)
 - considerations for IBM Spectrum Protect for Space Management [579](#)
- DMAPI file handle size considerations
 - for IBM Spectrum Protect for Space Management [579](#)
- documentation
 - installing man pages on AIX nodes [455](#)
 - installing man pages on Linux nodes [358](#)
- domain
 - Active Directory [466](#)
 - Windows [466](#)

E

- ECKD devices, preparing environment for [249](#)
- electronic license agreement
 - AIX nodes [454](#)
 - Linux nodes [355](#)
- Enable gpfs leases [311](#)
- enable leases [311](#)
- enable share modes [311](#)
- enabling
 - protocols [372](#)
- enabling file system features [267](#)
- enabling protocols
 - authentication considerations [293](#)
 - existing cluster [444](#)
- environment for ECKD devices, preparing [249](#)
- environment, preparing [354](#)
- ESS
 - adding protocol nodes [445](#)
 - deploying protocols [445](#)
- ESS 3000 awareness [429](#)
- ESS awareness [429](#)
- estimated node count [264](#)
- event based uploads
 - FTDC2CallHome [192](#)
- events
 - file audit logging [180](#)
- exceptions
 - CES NFS Linux [304](#)
- expiring a disconnected RO cache [78](#)
- explanations and examples, installing GPFS and protocols [407](#), [408](#)
- Express Edition
 - migrating to Standard [571](#)

- Extended attributes (EAs)
 - planning [283](#)
- external Kafka sink [184](#)
- extracting GPFS patches [358](#)
- extracting the IBM Spectrum Scale software [355](#)
- extracting update SUSE Linux Enterprise Server and Red Hat Enterprise Linux RPMs [358](#)
- extraction, packaging overview and [355](#)

F

- Fail-over scenarios [306](#)
- failback
 - new primary site [105](#)
 - old primary site [104](#)
- failed fresh upgrade
 - upgrade rerun [560](#)
- failover
 - secondary site [104](#)
- failover in AFM [80](#)
- failure
 - disk [226](#)
 - Network Shared Disk server [226](#)
 - node [223–225](#)
- failure groups
 - definition of [2](#)
 - loss of [248](#)
 - preventing loss of data access [226](#)
 - use of [247](#)
- features
 - AFM DR [99](#)
- file audit logging
 - attributes [181](#)
 - events [181](#)
 - fileset [179](#)
 - installing [489](#)
 - limitations [489](#)
 - producers [179](#)
 - records [179](#)
 - requirements [489](#)
- file authentication [294](#)
- File client limitations [306](#)
- file name considerations
 - Windows [461](#)
- file system creation [262](#)
- file system descriptor
 - failure groups [247](#)
 - inaccessible [248](#)
 - quorum [247](#)
- file system features
 - enabling [267](#)
- file system level backups [277](#)
- file system manager
 - command processing [22](#)
 - description [9](#)
 - internal log file [262](#)
 - mount of a file system [18](#)
 - NSD creation considerations [246](#)
 - quota management function [10](#)
 - selection of [10](#)
 - token management function [9](#)
 - Windows drive letter [265](#)
- file system name considerations
 - Windows [460](#)

- file systems
 - administrative state of [4](#), [25](#)
 - AFM node [11](#)
 - authorization [262](#)
 - block size [258](#)
 - creating [251](#)
 - descriptor [11](#)
 - device name [257](#)
 - disk descriptor [257](#)
 - enabling DMAPi [266](#)
 - interacting with a GPFS file system [18](#)
 - internal log file [262](#)
 - last time accessed [260](#)
 - list of disk descriptors [262](#)
 - maximum number of [12](#)
 - maximum number of files [267](#)
 - maximum number of mounted files [12](#)
 - maximum size [12](#)
 - maximum size supported [12](#)
 - metadata [11](#)
 - metadata integrity [11](#)
 - mount options [265](#)
 - mounting [3](#), [18](#), [257](#)
 - mountpoint [265](#)
 - NFS export [304](#)
 - NFS V4 export [304](#)
 - number of nodes mounted by [264](#)
 - opening a file [19](#)
 - quotas [265](#)
 - reading a file [19](#)
 - recoverability parameters [263](#), [264](#)
 - repairing [22](#)
 - sample creation [268](#)
 - shared access among clusters [1](#)
 - simultaneous access [2](#)
 - sizing [251](#)
 - stripe group [11](#)
 - time last modified [261](#)
 - Windows drive letter [265](#)
 - writing to a file [20](#), [21](#)
- file systems (controlling the order in which they are mounted) [268](#)
- files
 - .toc [455](#)
 - /etc/filesystems [25](#)
 - /etc/fstab [25](#)
 - /var/mmfs/etc/mmfs.cfg [25](#)
 - /var/mmfs/gen/mmsdrfs [25](#)
 - consistency of data [2](#)
 - GPFS recovery logs [13](#)
 - inode [12](#)
 - installation on AIX nodes [453](#)
 - maximum number of [12](#), [267](#)
 - mmfslinux [6](#)
 - structure within GPFS [11](#)
- files that can be created, maximum number of [267](#)
- fileset backups [277](#)
- fileset disabling
 - AFM [89](#)
- fileset to home
 - AFM [52](#)
- filesetdf option [267](#)
- filesets [3](#)
- firewall

firewall (*continued*)

- Windows, disabling [466](#)
- firewall recommendations
 - transparent cloud tiering [334](#)
- Force flushing contents before Async Delay [59](#)
- fresh transparent cloud tiering cluster
 - setting up [503](#)
- fresh upgrade failure
 - solution [560](#)
- Functional overview
 - REST API [150](#)

G

Ganesha limitations [304](#)

gateway

- node failure [71](#)
- recovery [71](#)

GDS [26](#), [250](#), [481](#), [517](#)

global namespace

- AFM [45](#)

Google Cloud Platform [139](#)

GPFS

- adding file systems [443](#)
- adding LTFS nodes [446](#)
- adding nodes [443](#), [445](#), [446](#)
- adding nodes with toolkit [442](#)
- adding NSDs [443](#)
- adding protocol nodes [445](#)
- adding protocols with toolkit [442](#)
- administration commands [4](#)
- AFM gateway node [11](#)
- application interaction [18–22](#)
- architecture [8](#), [11](#), [14](#), [15](#), [18](#)
- backup data [26](#)
- basic structure [4–6](#)
- CES [29](#)
- CES NFS support [31](#)
- CES node [11](#)
- cluster configuration data files [25](#)
- cluster configurations [6](#)
- cluster creation [228–230](#), [232–237](#)
- Cluster Export Services [29](#)
- cluster manager [9](#)
- considerations for applications [346](#)
- daemon [5](#)
- daemon communication [16](#)
- deploying protocols [416](#), [420](#), [421](#), [426](#), [427](#), [442](#), [543](#), [545](#), [552](#)
- diagnosing errors [444](#)
- disk considerations [239](#), [246–249](#)
- disk storage use [11](#)
- enabling protocols [444](#)
- enabling protocols on Linux [372](#)
- Express Edition [571](#)
- failure recovery processing [24](#)
- file consistency [2](#)
- file structure [11](#), [13](#)
- file system creation [251](#), [257](#), [258](#), [260–268](#)
- file system manager [9](#)
- for improved performance [2](#)
- hardware requirements [219](#)
- increased data availability [2](#)

GPFS (*continued*)

- installing [349](#), [351](#), [352](#), [354](#), [355](#), [357–359](#), [364](#), [365](#), [373](#), [374](#), [376](#), [379](#), [386](#), [392](#), [394](#), [407–409](#), [414](#), [416](#), [420](#), [421](#), [426](#), [427](#), [446](#), [448](#), [449](#), [451](#), [453–455](#), [457–466](#), [468–470](#), [474](#), [476](#), [527](#), [543](#), [552](#)
- installing CES on Linux [371](#)
- installing on AIX nodes [453–455](#)
- installing on Linux [366](#)
- installing on Linux nodes [351](#), [352](#), [354](#), [355](#), [358](#), [359](#), [373](#), [374](#), [376](#), [379](#), [386](#), [392](#), [407–409](#), [414](#), [446](#), [448](#), [449](#), [451](#), [527](#)
- installing on Windows nodes [457–466](#), [468–470](#), [474](#)
- installing packages on Linux [367](#), [370](#)
- installing protocols on Linux [371](#)
- installing with toolkit [386](#)
- IP address usage [16](#)
- kernel extensions [5](#)
- management functions [9](#), [11](#)
- memory usage [14](#), [15](#)
- metanode [11](#)
- migration [505](#), [577–579](#)
- multi-region object deployment [35](#), [326–329](#)
- network communication [15–17](#)
- NSD disk discovery [23](#)
- object capabilities [37](#)
- object storage support [33–37](#)
- OS upgrade [580](#), [585](#)
- pinned memory [14](#)
- planning [215](#), [219](#), [220](#), [223](#), [225](#), [226](#), [228](#), [251](#), [270–272](#), [275](#), [277](#), [279](#), [281](#), [291](#), [293](#), [294](#), [299](#), [301–303](#), [315](#), [317](#), [319–324](#)
- planning for transparent cloud tiering [331](#)
- portability layer [6](#)
- product editions [213](#), [218](#)
- product overview [168](#)
- protocol node [11](#)
- protocols support [28](#), [29](#), [31–33](#)
- quota files [13](#)
- recoverability [223–226](#), [228](#)
- recovery logs [13](#)
- RHEL 8.x upgrade [583](#)
- S3 API [36](#)
- servicing protocol nodes [589](#)
- simplified administration [4](#)
- simplified storage management [3](#)
- SMB support [32](#)
- software requirements [220](#)
- special management functions [8](#)
- strengths [1–4](#)
- stretch cluster use case [431](#)
- system flexibility [3](#)
- thin provisioned disks [240](#)
- unified file and object access [34](#), [325](#), [326](#)
- uninstalling [495](#), [496](#), [502](#)
- upgrade [508](#), [509](#), [512](#), [572](#)
- upgrading [505–509](#), [512](#), [520](#), [521](#), [523](#), [526](#), [527](#), [545](#), [572](#), [577–580](#), [586](#)
- upgrading from 3.5 [590](#)
- user interaction [18–22](#)
- GPFS administration commands [15](#)
- GPFS administrative adapter port name [229](#)
- GPFS architecture [8](#)
- GPFS clusters
 - administration adapter port name [229](#)

- GPFS clusters (*continued*)
 - configuration data files [4](#)
 - configuration file [237](#)
 - configuration servers [232](#)
 - creating [228](#)
 - daemon
 - starting [236](#)
 - daemon communication [16](#)
 - establishing [349](#)
 - introduction [1](#)
 - naming [235](#)
 - nodes in the cluster [230](#), [232–237](#)
 - operating environment [6](#)
 - planning nodes [228](#)
 - portability layer [364](#), [365](#)
 - recovery logs
 - creation of [13](#)
 - unavailable [22](#)
 - server nodes [228](#)
 - starting [349](#)
 - starting the GPFS daemon [9](#), [236](#)
 - user ID domain [235](#)
- GPFS communications adapter port name [229](#)
- GPFS daemon communications [15](#)
- GPFS for Linux on Z, running [358](#)
- GPFS for Windows Multiplatform
 - overview [457](#)
- GPFS introduction [1](#)
- GPFS limitations on Windows [459](#)
- GPFS patches, extracting [358](#)
- GPFS product structure
 - capacity based licensing [218](#)
- GPFS strengths [1](#)
- GPFS, configuring [409](#)
- GPFS, installing [392](#)
- GPFS, installing over a network [455](#)
- GPFS, planning for [219](#)
- GPFS, reverting to a previous level [578](#)
- GPG key
 - verifying [357](#)
- GPU [26](#)
- GPUDirect Storage
 - installing [481](#)
 - planning [250](#)
 - upgrading [517](#)
- GSKit [1](#)
- GUI
 - overview [145](#)
- GUI without root privileges [385](#)
- guidelines for maintenance tasks [339](#)

H

- hard limit, quotas [265](#)
- hardware requirements [219](#)
- Hardware requirements
 - transparent cloud tiering [329](#)
- heartbeats
 - call home events [194](#)
 - mmhealth command [194](#)
- highly available write cache
 - planning [286](#)
- home
 - AFM [39](#)

- home cluster
 - UID/GID requirements [343](#)
- How WORM works [174](#)
- HPC server (Windows), configuring [474](#)

I

- IBM Cloud Object Storage migration
 - transparent cloud tiering [541](#)
- IBM Cloud Object Storage considerations [332](#)
- IBM Global Security Kit (GSKit) [1](#)
- IBM Spectrum Archive
 - adding nodes [446](#)
- IBM Spectrum Protect
 - backup planning [270–272](#), [275](#), [277](#), [279](#), [281](#)
 - file data storage [271](#)
 - fileset backups [279](#)
 - identify backup candidates [272](#)
 - metadata storage [271](#)
 - provisioning [270](#)
- IBM Spectrum Protect backup planning [270](#), [272](#)
- IBM Spectrum Protect for Space Management
 - DMAPI file handle size considerations [579](#)
 - file system backups [281](#)
- IBM Spectrum Protect space managed file system backups [281](#)
- IBM Spectrum Scale
 - 5.0.y [508](#)
 - 5.1.x [508](#)
 - Active File Management [39–41](#), [43](#), [45–48](#), [52](#), [53](#), [56](#), [59](#), [61](#), [63](#), [70](#), [71](#), [73](#), [77–80](#), [82](#), [85](#), [87](#), [89](#), [117](#), [129](#), [341–343](#), [483](#), [518](#)
 - Active File Management - Disaster Recovery [343](#)
 - Active File Management DR [97–100](#), [104–106](#), [343](#), [344](#), [485](#)
 - adding file systems [443](#)
 - adding nodes [443](#), [445](#), [446](#)
 - adding nodes with toolkit [442](#)
 - adding NSDs [443](#)
 - adding protocols with toolkit [442](#)
 - administration commands [4](#)
- AFM
 - .afm [53](#)
 - .pconflicts [53](#)
 - .ptrash [53](#)
 - AFM gateway nodes requirements on the cache and home clusters [342](#)
 - AFM to Cloud Object Server replication
 - Eviction feature [129](#)
 - AFM to cloud object storage [117](#)
 - asynchronous delay [48](#)
 - cache cluster [39](#)
 - cache eviction [73](#)
 - cached and uncached files [47](#)
 - caching modes [40](#)
 - encryption [89](#)
 - expiring a disconnected RO cache [78](#)
 - Failover [80](#)
 - Fast creates [56](#)
 - fileset disabling [89](#)
 - fileset to home [52](#)
 - Force flushing contents before Async Delay [59](#)
 - gateway node [343](#)

IBM Spectrum Scale (continued)

AFM (continued)

- gateway node failure [71](#)
- gateway recovery [71](#)
- global namespace [45](#)
- home [39](#)
- IBM Spectrum Protect [85](#)
- Inode limits to set at cache and home [342](#)
- install [483](#)
- iw cache maintenance [87](#)
- NFS backend protocol [41](#)
- operation with disconnected home [77](#)
- Parallel data transfer using multiple remote mounts [61](#)
- Parallel data transfers [59](#)
- partial file caching [63](#)
- peer snapshot [70](#)
- planned maintenance [87](#)
- planning [341](#)
- prefetch [63](#)
- primary gateway [43](#)
- psnap [70](#)
- resync on SW filesets [82](#)
- revalidation parameters [46](#)
- synchronous or asynchronous operations [48](#)
- UID and GID requirements on the cache and home clusters [341](#)
- unplanned maintenance [87](#)
- upgrade [518](#)
- using mmbbackup [89](#)
- viewing snapshots at home [79](#)
- WAN latency [39](#)
- workerThreads on cache cluster [342](#)

- AFM communication
 - cache and home [41](#)

AFM DR

- concepts [99](#)
- failback [104](#), [105](#)
- failover [104](#)
- installing [485](#)
- introduction [97](#)
- modes [99](#)
- planning [343](#)
- recovery time objective [98](#)
- role reversal [102](#)
- RPO snapshots [100](#)
- secondary cluster [344](#)
- secondary site [105](#), [106](#)
- UID/GID requirements [343](#)
- worker1Threads on primary cluster [343](#)

- AFM inode limits [342](#)

AFM mode

- operations [48](#)

AFM monitoring

- cache and home [41](#)

- AFM-based Asynchronous Disaster Recovery
 - role reversal [102](#)

- application interaction [18–22](#)

- architecture [8](#), [11](#), [14](#), [15](#), [18](#)

- backup data [26](#)

- basic structure [4–6](#)

- call home [191](#), [194](#), [208](#), [209](#), [487](#)

- call home data upload [191](#), [194](#), [208](#)

- call home update PTF [209](#)

IBM Spectrum Scale (continued)

- capacity based licensing [218](#)

- CES [29](#)

- CES HDFS planning [329](#)

- CES NFS support [31](#)

- cluster configuration data files [25](#)

- cluster configurations [6](#)

- cluster creation [228–230](#), [232–237](#)

- Cluster Export Services [29](#)

- cluster manager [9](#)

- configuring and tuning system

- transparent cloud tiering [332](#)

- considerations for applications [346](#)

- daemon [5](#)

- daemon communication [16](#)

- data protection [144](#)

- deploying protocols [416](#), [420](#), [421](#), [426](#), [427](#), [442](#), [543](#), [552](#)

- diagnosing errors [444](#)

- disk considerations [238](#), [239](#), [246–249](#)

- enabling protocols [444](#)

- enabling protocols on Linux [372](#)

- Express Edition [571](#)

- failure recovery processing [24](#)

- file audit logging [489](#)

- file consistency [2](#)

- file structure [11](#), [13](#)

- file system creation [251](#), [257](#), [258](#), [260–268](#)

- for improved performance [2](#)

- GUI [145](#)

- hardware requirements [219](#)

- Immutability and appendOnly

- IAM modes [99](#)

- increased data availability [2](#)

- inspect call home [206](#)

- inspect call home data upload [206](#)

- installation toolkit [402](#), [404](#), [422](#), [424](#), [429](#)

- installing [349](#), [351](#), [352](#), [354](#), [355](#), [357–359](#), [364](#), [365](#), [373](#), [374](#), [376](#), [379](#), [386](#), [392](#), [394](#), [407–409](#), [414](#), [416](#), [420](#), [421](#), [426](#), [427](#), [446](#), [448](#), [449](#), [451](#), [453–455](#), [457–466](#), [468–470](#), [474](#), [476](#), [527](#), [543](#), [552](#)

- installing call home [487](#)

- installing CES on Linux [371](#)

- installing GUI [379](#)

- installing on AIX nodes [453–455](#)

- installing on Linux [366](#)

- installing on Linux nodes [351](#), [352](#), [354](#), [355](#), [357–359](#), [373](#), [374](#), [376](#), [379](#), [386](#), [392](#), [407–409](#), [414](#), [446](#), [448](#), [449](#), [451](#), [527](#)

- installing on Windows nodes [457–466](#), [468–470](#), [474](#)

- installing packages on Linux [367](#), [370](#)

- installing protocols on Linux [371](#)

- installing with toolkit [386](#)

- IP address usage [16](#)

- kernel extensions [5](#)

- key features [1](#)

- licensing [213](#)

- management functions [9](#), [11](#)

- management GUI

- manual installation [379](#)

- node classes [503](#)

- manual installation [359](#), [373](#), [374](#), [376](#), [527](#)

- manual upgrade [526](#), [527](#)

- memory usage [14](#), [15](#)

IBM Spectrum Scale *(continued)*

- migration [505](#), [520](#), [521](#), [523](#), [577–579](#)
- multi-region object deployment [35](#), [326–329](#)
- network communication [15–17](#)
- node failure [223–225](#)
- non-protocol nodes [509](#)
- NSD disk discovery [23](#)
- object capabilities [37](#)
- object storage planning [317](#), [319–324](#)
- object storage support [33–37](#)
- offline upgrade [554](#)
- OpenStack cloud deployment [210](#)
- OS upgrade [580](#), [585](#)
- pinned memory [14](#)
- planning [215](#), [219](#), [220](#), [223](#), [225](#), [226](#), [228](#), [238](#), [251](#), [270–272](#), [275](#), [277](#), [279](#), [281](#), [291](#), [293](#), [294](#), [299](#), [301–303](#), [315](#), [317](#), [320–324](#), [335](#)
- planning for transparent cloud tiering [331](#)
- portability layer [6](#)
- prerequisites for call home [487](#)
- product edition [566](#)
- product editions [213](#), [218](#)
- product overview [168](#)
- product structure [213](#)
- protocol nodes [512](#)
- protocol nodes authentication [563](#)
- protocols prerequisites [352](#)
- protocols support [28](#), [29](#), [31–33](#), [329](#)
- recoverability [223–226](#), [228](#)
- RHEL 8.x upgrade [583](#)
- S3 API [36](#)
- SELinux
 - enforcing mode [346](#)
 - permissive mode [346](#)
- servicing protocol nodes [589](#)
- shared file system access [1](#)
- simplified administration [4](#)
- simplified storage management [3](#)
- SMB support [32](#)
- software requirements [220](#)
- special management functions [8](#)
- strengths [1–4](#)
- stretch cluster use case [431](#)
- supported cloud providers [175](#)
- supported NFS clients [303](#)
- supported upgrade paths [506](#)
- system flexibility [3](#)
- thin provisioned disks [240](#)
- transparent cloud tiering
 - how it works [170](#)
 - overview [168](#)
- unified file and object access [34](#), [325](#), [326](#)
- uninstalling [495](#), [496](#), [502](#)
- upgrade
 - non-protocol nodes [509](#)
 - protocol nodes [512](#)
- upgrade considerations [507](#)
- upgrade process flow [545](#)
- upgrading
 - non-protocol nodes [509](#)
 - protocol nodes [512](#)
 - protocol nodes authentication [563](#)
- Upgrading [508](#), [509](#), [512](#), [563](#)
- upgrading from 3.5 [590](#)

IBM Spectrum Scale *(continued)*

- upgrading from unsupported OS [586](#)
- upgrading LTFS nodes [562](#)
- user interaction [18–22](#)
- WORM solutions [174](#)
- IBM Spectrum Scale AFM node [11](#)
- IBM Spectrum Scale CES node [11](#)
- IBM Spectrum Scale Client license [215](#)
- IBM Spectrum Scale disk storage use [11](#)
- IBM Spectrum Scale file system
 - migrating SMB data [308](#)
- IBM Spectrum Scale file system manager [9](#)
- IBM Spectrum Scale for object storage
 - manual install [374](#)
 - multi-region object deployment [35](#), [326–329](#)
 - object capabilities [37](#)
 - S3 API [36](#)
 - storage policies [34](#)
 - unified file and object access [34](#), [325](#), [326](#)
 - upgrade to 5.0.x [520](#)
 - upgrading [520](#)
- IBM Spectrum Scale FPO license [215](#)
- IBM Spectrum Scale GUI
 - upgrading to the latest version [529](#)
- IBM Spectrum Scale information units [xv](#)
- IBM Spectrum Scale introduction [1](#)
- IBM Spectrum Scale license designation [215](#)
- IBM Spectrum Scale management API
 - configuring [479](#)
 - fields parameter [152](#)
 - filter parameter [154](#)
 - Functional overview [150](#)
 - installing [479](#)
 - paging [157](#)
- IBM Spectrum Scale metanode [11](#)
- IBM Spectrum Scale NFS
 - upgrading [523](#)
- IBM Spectrum Scale object storage
 - overview [33](#)
- IBM Spectrum Scale overview [1](#)
- IBM Spectrum Scale protocol node [11](#)
- IBM Spectrum Scale protocols
 - planning [291](#), [293](#), [294](#), [299](#), [301–303](#), [317](#), [320–329](#)
- IBM Spectrum Scale quota files [13](#)
- IBM Spectrum Scale recovery logs [13](#)
- IBM Spectrum Scale Server license [215](#)
- IBM Z, DASD tested with Linux on [248](#)
- IBM Z, running GPFS for Linux on [358](#)
- IDMU (Identity Mapping for Unix (IDMU) / RFC 2307 Attributes) [470](#)
- Independent writer
 - AFM mode [48](#)
- independent-writer (IW) [40](#)
- indirect blocks [11](#), [13](#)
- indirection level [11](#)
- initialization of the GPFS daemon [18](#)
- inode
 - allocation file [12](#)
 - allocation map [13](#)
 - cache [14](#)
 - logging of [13](#)
 - usage [11](#), [21](#)
- Inode limits to set at cache and home [342](#)
- install

- install (*continued*)
 - query
 - uninstall [476](#)
- installation cleanup procedures [496](#)
- installation toolkit
 - audit logging [422](#)
 - call home configuration [424](#)
 - ESS 3000 awareness [429](#)
 - ESS awareness [429](#)
 - limitations [394](#), [427](#)
 - mixed operating system support [402](#)
 - populating cluster definition file [427](#)
 - preparing [404](#)
 - product edition change [566](#)
 - stretch cluster [431](#)
 - upgrade flow chart [545](#)
 - upgrade process flow [545](#)
 - upgrading LTFS nodes [562](#)
- installation toolkit(
 - offline upgrade [554](#)
- installing
 - AFM-based disaster recovery [485](#)
 - api [479](#)
 - CES [371](#)
 - cloud data sharing [476](#)
 - Cloud services [476](#)
 - clustered watch folder [491](#)
 - GPUDirect Storage [481](#)
 - manually [366](#)
 - transparent cloud tiering
 - on GPFS nodes [476](#)
- installing and configuring OpenSSH, Windows [472](#)
- installing Cygwin [466](#)
- installing GPFS
 - on Windows nodes [468](#), [469](#)
 - over a network [455](#)
- installing GPFS (Ubuntu)
 - verifying the GPFS installation [373](#)
- installing GPFS and protocols, examples [407](#), [408](#), [416](#), [421](#)
- installing GPFS on AIX nodes
 - creating the GPFS directory [454](#)
 - directions [455](#)
 - electronic license agreement [454](#)
 - files used during [453](#)
 - man pages [455](#)
 - procedures for [453](#)
 - table of contents file [455](#)
 - verifying the GPFS installation [455](#)
 - what to do before you install GPFS [453](#)
- installing GPFS on Linux nodes
 - building the GPFS portability layer
 - using the Autoconfig tool [365](#)
 - using the mmbuildgpl command [364](#), [365](#)
 - electronic license agreement [355](#)
 - installing the software packages [359](#), [373](#), [374](#), [376](#), [379](#), [527](#)
 - man pages [358](#)
 - procedure for [351](#), [352](#), [355](#), [358](#), [359](#), [373](#), [374](#), [376](#), [379](#), [386](#), [392](#), [407–409](#), [414](#), [446](#), [448](#), [449](#), [451](#), [527](#)
 - verifying the GPFS installation [374](#)
 - what to do before you install GPFS [351](#)
- installing GPFS on Windows nodes [457](#)
- installing GPFS prerequisites [465](#)
- installing IBM Spectrum Scale on Linux nodes

- installing IBM Spectrum Scale on Linux nodes (*continued*)
 - creating the GPFS directory [355](#)
 - License Acceptance Process (LAP) tool [355](#)
- installing packages
 - manually [367](#)
 - NSDs [370](#)
 - with toolkit [386](#)
- installing protocols [351](#), [352](#), [355](#), [386](#), [426](#)
- installing Tracefmt [466](#)
- installing Tracelog [466](#)
- inter-protocol level locking [311](#)
- Internet Protocol Version 6 [91](#), [92](#)
- interoperability of
 - transparent cloud tiering [175](#)
- introduction
 - AFM DR [97](#)
 - file audit logging [178](#)
 - transparent cloud tiering [168](#)
- invariant address adapter
 - requirement [219](#)
- IO Throughput [310](#)
- IOPS [310](#)
- IP address
 - private [16](#)
 - public [16](#)

J

- joining an Active Directory domain [466](#)
- JSON
 - attributes [181](#)
 - events [181](#)
- JSON attributes [186](#)

K

- Kafka
 - external sink [184](#)
- kernel extensions [5](#)
- kernel memory [14](#)

L

- latest level of file system
 - migrating [266](#)
- librdkafka [542](#)
- License Acceptance Process (LAP) tool [355](#)
- license designation [215](#)
- limitations
 - CES NFS Linux [304](#)
 - clustered watch folder [491](#)
 - NFS protocol nodes [304](#)
- Limitations
 - SMB clients limitations [306](#)
 - SMB share limitations [306](#)
- limitations of
 - transparent cloud tiering [175](#)
- Linux
 - building the GPFS portability layer
 - using the mmbuildgpl command [364](#), [365](#)
 - installation instructions for GPFS [351](#)
 - installing GPFS [355](#)
 - kernel requirement [220](#)

Linux (*continued*)

prerequisite software [354](#)

Linux on Z, DASD tested with [248](#)

Linux on Z, running GPFS [358](#)

load balancing across disks [2](#)

Local updates

AFM mode [48](#)

local-update (LU) [40](#)

M

maintenance

iw cache [87](#)

maintenance levels of GPFS, applying [580](#)

maintenance tasks

planning [339](#)

man pages

installing on AIX nodes [455](#)

installing on Linux nodes [358](#)

management API

fields parameter [152](#)

filter parameter [154](#)

paging [157](#)

management GUI

installing [426](#)

node classes [503](#)

nodeclasses [379](#)

uninstalling [502](#)

management GUI node classes

removing nodes [503](#)

manual installation [359](#), [366](#), [367](#), [370–374](#), [376](#), [527](#)

manual upgrade

performance monitoring

Ubuntu 18.04 [527](#)

maxFilesToCache parameter

memory usage [15](#)

maximum data replicas [264](#)

maximum metadata replicas [263](#)

maximum number of files [267](#)

maximum number of files that can be created [267](#)

maxStatCache parameter

memory usage [15](#)

memory

non-pinned [15](#)

pinned [14](#)

usage [14](#)

metadata

default [263](#)

maximum replicas [263](#)

replication [263](#)

metanode [11](#)

migrating

reverting to the previous level of GPFS [577](#), [578](#)

migrating file system format to the latest level [266](#)

migrating IBM Cloud Object Storage [541](#)

Migrating SMB data from traditional filer [308](#)

migrating SMB data from traditional NAS [308](#)

migrating SMB data to IBM Spectrum Scale [308](#)

mixed Windows and UNIX cluster, configuring [470](#)

mmbackup command [26](#), [144](#)

mmbackupconfig command [144](#)

mmcesdr command [144](#)

mmchcluster command [233](#), [234](#)

mmchconfig command

mmchconfig command (*continued*)

defining a subset of nodes [4](#), [17](#)

mmchdisk command [22](#)

mmcheckquota command [18](#)

mmchfs [304](#)

mmchfs command [251](#), [262](#), [265](#)

mmcrcluster command [16](#), [228](#), [233](#), [234](#), [453](#)

mmcrfs [304](#)

mmcrfs command [251](#), [262](#), [265](#)

mmcrnsd command [239](#)

mmdefedquota command [265](#), [266](#)

mmdefquotaon command [266](#)

mmdelnsd command [239](#)

mmedquota command [265](#)

mmfsck command [13](#), [18](#), [22](#), [144](#)

mmimagebackup command [144](#)

mmimgrestore command [144](#)

mmlsdisk command [22](#), [248](#)

mmlsfs [304](#)

mmlsfs command [13](#)

mmlsquota command [265](#)

mmmount command [18](#)

mmrepquota command [265](#)

mmrestoreconfig command [144](#)

mmrestorefs command [144](#)

mmstartup command [236](#)

mmwinservctl command [233](#), [234](#)

modes

AFM DR [99](#)

monitoring

cache and home [41](#)

mount options [265](#)

mount-priority option [268](#)

mounting a file system [18](#), [257](#), [265](#)

mounting of file systems, controlling the order of the [268](#)

mountpoint [265](#)

moving SMB data from a traditional filer [308](#)

mtime values [261](#)

multi-node Cloud services

set up [477](#)

multi-node setup for Cloud services [477](#)

multi-region object deployment

authentication planning [327](#)

data protection planning [328](#)

enabling [448](#), [449](#)

Keystone endpoints [327](#)

monitoring planning [329](#)

network planning [328](#)

overview [35](#)

performance considerations [329](#)

planning [326–328](#)

Multiple Path I/O (MPIO)

utilizing [227](#)

N

NAS protocols [311](#)

network

communication within your cluster [2](#)

network communication

administration commands [17](#)

network considerations [331](#)

Network File System (NFS)

access control lists [262](#)

- Network File System (NFS) (*continued*)
 - deny-write open lock [257](#)
- network installing GPFS [455](#)
- Network Shared Disk (NSD)
 - creation of [239](#)
 - disk discovery [23](#)
 - server disk considerations [238](#)
 - server failure [226](#)
 - server node considerations [246](#)
- NFS
 - backend protocol [41](#)
 - upgrade to 5.0.x [523](#)
- NFS planning
 - file system considerations [302](#)
 - NFS client considerations [303](#)
- NFS-supported clients [303](#)
- node quorum
 - definition of [224](#)
 - selecting nodes [225](#)
- node quorum with tiebreaker disks
 - definition of [224](#)
 - selecting nodes [225](#)
- nodes
 - acting as special managers [8](#)
 - cluster manager [9](#)
 - descriptor form [230](#)
 - designation as manager or client [231](#)
 - estimating the number of [264](#)
 - failure [223–225](#)
 - file of nodes in the cluster for installation [453](#)
 - file system manager [9](#)
 - file system manager selection [10](#)
 - in a GPFS cluster [228](#)
 - in the GPFS cluster [230](#)
 - quorum [231](#)
- nofilesetdf option [267](#)
- non-pinned memory [15](#)
- NSD
 - backend protocol [41](#)
- number of files that can be created, maximum [267](#)
- NVIDIA [26](#)

O

- Object
 - upgrade to 5.0.x [520](#)
- object authentication [299](#)
- object capabilities [37](#)
- object heatmap data tiering [38](#)
- object storage
 - overview [33](#)
- object storage planning
 - authentication method [322](#)
 - backup [322](#)
 - cluster host name [321](#)
 - disaster recovery [322](#)
 - load balancing [320](#)
 - OpenStack repo [319](#)
 - SELinux considerations [323, 324](#)
- object support
 - overview [33](#)
- Open Secure Socket Layer [1](#)
- OpenSSL [1](#)

- OpenStack repo [319](#)
- operating system
 - calls [19–21](#)
 - commands [18](#)
- opportunistic locks [311](#)
- order in which file systems are mounted, controlling the [268](#)
- overview
 - of GPFS for Windows Multiplatform [457](#)
 - transparent cloud tiering [168](#)

P

- packaging overview and extraction [355](#)
- pagepool parameter
 - affect on performance [20](#)
 - in support of I/O [14](#)
 - memory usage [14](#)
- Parallel data transfer [93](#)
- Parallel data transfer using multiple remote mounts [61](#)
- Parallel data transfers [59](#)
- partial file caching [63](#)
- patches (GPFS), extracting [358](#)
- patches (GPFS), extracting SUSE Linux Enterprise Server and Red Hat Enterprise Linux [358](#)
- PATH environment variable [453](#)
- peer snapshot [70](#)
- performance
 - pagepool parameter [20](#)
 - use of GPFS to improve [2](#)
 - use of pagepool [14](#)
- performance considerations
 - Cloud services [335](#)
 - transparent cloud tiering [335](#)
- performance monitoring
 - mmperfmon command [142](#)
 - performance monitoring tool [142](#)
- performance monitoring tool
 - manual installation [376](#)
- Performance Monitoring tool
 - manual installation [527](#)
 - pmswift [527](#)
 - uninstalling [502](#)
- Persistent Reserve
 - reduced recovery time [228](#)
- pinned memory [14](#)
- planned maintenance
 - IW cache [87](#)
- planning
 - GPUDirect Storage [250](#)
 - highly available write cache [286](#)
 - IBM Spectrum Scale [338](#)
 - NFS [301–303](#)
 - object storage [317, 320–329](#)
 - SMB
 - SMB fail-over scenarios [306](#)
 - SMB upgrades [306](#)
 - systemd [289](#)
 - transparent cloud tiering [329](#)
- Planning
 - extended attributes (EAs) [283](#)
 - Quality of Service for I/O operations (QoS) [282](#)
- planning considerations
 - cluster creation [228](#)

- planning considerations (*continued*)
 - disks [238](#)
 - file system creation [251](#)
 - hardware requirements [219](#)
 - IBM Spectrum Scale license designation [215](#)
 - product structure [213](#), [218](#)
 - recoverability [223](#)
 - software requirements [220](#)
- planning for
 - cloud data sharing [329](#)
 - IBM Spectrum Scale [335](#)
- planning for IBM Spectrum Scale [329](#)
- planning for maintenance activities
 - Cloud services [339](#)
- policies [3](#)
- populating cluster definition file
 - limitations [427](#)
- port forwarding [385](#)
- portability layer
 - building
 - using the mmbuildgpl command [364](#), [365](#)
 - description [6](#)
- Posix locking [311](#)
- pre-installation steps
 - transparent cloud tiering [475](#)
- preparing direct access storage devices (DASD) for NSDs [248](#)
- preparing environment for ECKD devices [249](#)
- preparing the environment [354](#)
- prerequisites
 - for Windows [465](#)
- prerequisites for installing protocols [352](#), [355](#)
- primary gateway [43](#)
- private IP address [16](#)
- producers
 - clustered watch folder [184](#)
 - lightweight events [179](#)
- product edition
 - changing [566](#)
- programming specifications
 - AIX prerequisite software [453](#)
 - Linux prerequisite software [354](#)
 - verifying prerequisite software [354](#), [453](#)
- protocol exports
 - fileset considerations [315](#)
- protocol node configuration
 - object protocol [448](#), [449](#), [451](#)
- protocol node hardware [310](#)
- protocol nodes
 - driver updates [589](#)
 - firmware updates [589](#)
 - kernel updates [589](#)
 - servicing [589](#)
 - upgrading [509](#), [512](#)
- protocol nodes authentication
 - upgrading [563](#)
- protocols prerequisites [352](#), [355](#)
- protocols support
 - overview [28](#)
- protocols, deploying [416](#), [421](#)
- protocols, installing [351](#), [426](#)
- psnap [70](#)
- PTF support [580](#)
- public IP address [16](#)

Q

- Quality of Service for I/O operations (QoS)
 - planning [282](#)
- quorum
 - definition of [224](#)
 - during node failure [223](#), [224](#)
 - enforcement [9](#)
 - file system descriptor [247](#)
 - initialization of GPFS [18](#)
 - node [224](#)
 - selecting nodes [225](#)
- quota support
 - Cloud services [341](#)
- quotas
 - default quotas [266](#)
 - description [265](#)
 - files [13](#)
 - in a replicated system [265](#)
 - role of file system manager node [10](#)
 - values reported in a replicated file system [265](#)

R

- rcp command [234](#)
- Read only
 - AFM mode [48](#)
- read operation
 - buffer available [20](#)
 - buffer not available [20](#)
 - requirements [19](#)
 - token management [20](#)
- read-only (RO) [40](#)
- recommendations for maintenance tasks
 - Cloud services [339](#)
- recoverability
 - disk failure [226](#)
 - disks [22](#)
 - features of GPFS [2](#), [24](#)
 - file systems [22](#)
 - node failure [223](#)
 - parameters [223](#)
- recovery time
 - reducing with Persistent Reserve [228](#)
- recovery time objective
 - AFM DR [98](#)
- Red Hat Enterprise Linux 8.x
 - removing object protocol [585](#)
- reduced recovery time using Persistent Reserve [228](#)
- Redundant Array of Independent Disks (RAID)
 - preventing loss of data access [226](#)
- remote
 - file systems [183](#)
- remote command environment
 - rcp [234](#)
 - rsh [233](#)
 - scp [234](#)
 - ssh [233](#)
- remotely
 - mounted [183](#)
- remotely mounted
 - file system [490](#)
- remotely mounted file systems [183](#)

- removing GPFS, uninstalling [495](#)
- repairing a file system [22](#)
- replication
 - affect on quotas [265](#)
 - description of [2](#)
 - preventing loss of data access [226](#)
- reporting numbers based on quotas for the fileset [267](#)
- Requests
 - REST API [151](#), [157](#), [158](#)
- requirements
 - clustered watch folder [491](#)
 - hardware [219](#), [489](#)
 - OS [489](#)
 - RPM [489](#)
 - software [220](#)
 - system [489](#)
- Response time [310](#)
- Responses
 - REST API [155](#)
- REST API
 - configuring [479](#)
 - Functional overview [150](#)
 - installing [479](#)
 - Requests [151](#), [157](#), [158](#)
 - Responses [155](#)
- resync on SW filesets [82](#)
- revalidation
 - AFM [46](#)
 - parameters [46](#)
- reverting to a previous level [577](#), [578](#)
- role reversal
 - AFM DR [102](#)
- RPM database
 - query
 - RPM commands [476](#)
- RPM package
 - covert
 - deb package [476](#)
- RPMs (update), extracting SUSE Linux Enterprise Server and Linux [358](#)
- RPO snapshots [100](#)
- rsh command [233](#)
- running GPFS for Linux on Z [358](#)

S

- S3 API
 - overview [36](#)
- scalegmt user [385](#)
- scp command [234](#)
- secondary cluster
 - NFS setup [344](#)
- secure communication
 - establishing [37](#)
- security
 - shared file system access [1](#)
- security considerations
 - transparent cloud tiering [338](#)
- self-extracting package [355](#)
- servicing protocol nodes [589](#)
- servicing your GPFS system [580](#)
- set up multi-nodes [477](#)
- setting quota limit
 - transparent cloud tiering [341](#)

- setting up a multi-node [477](#)
- share modes [311](#)
- shared file system access [1](#)
- shared segments [14](#)
- shell PATH [354](#)
- Single writer
 - AFM mode [48](#)
- single-writer (SW) [40](#)
- sizing file systems [251](#)
- SMB
 - planning [306](#)
 - update to 4.2.3.x [521](#)
 - update to 5.0.x [521](#)
 - upgrading [521](#), [523](#)
- SMB Client [311](#), [315](#)
- SMB connections
 - SMB active connections [305](#)
- SMB data migration [308](#)
- SMB data migration using Robocopy [308](#)
- SMB export [311](#)
- SMB migration prerequisites [308](#)
- SMB planning [305](#)
- SMB protocol [311](#)
- SMB share limitations [306](#)
- SMB support
 - overview [32](#)
- snapshot based backups [275](#)
- snapshots
 - coexistence considerations with DMAPi [579](#)
 - gpfs.snap command [140](#)
- SOBAR [144](#)
- soft limit, quotas [265](#)
- softcopy documentation [358](#), [455](#)
- software requirements [220](#)
- Software requirements
 - transparent cloud tiering [329](#)
- Specifying whether the df command will report numbers based on quotas for the fileset [267](#)
- spectrum scale API [149](#)
- spectrumscale [386](#)
- spectrumscale installation toolkit
 - adding file systems [443](#)
 - adding node definitions [409](#)
 - adding nodes [443](#)
 - adding NSD nodes [411](#)
 - adding NSDs [443](#)
 - configuration options [409](#)
 - creating file systems [412](#)
 - debugging [444](#)
 - deploying protocols
 - debugging [421](#)
 - logging [421](#)
 - diagnosing errors [444](#)
 - enabling protocols [444](#)
 - installing GPFS [414](#)
 - installing IBM Spectrum Scale [414](#), [416](#), [420](#), [421](#)
 - installing management GUI [426](#)
 - limitations [392](#), [394](#), [427](#)
 - options [392](#), [394](#)
 - overview [386](#)
 - populating cluster definition file [426](#), [427](#)
 - preparing [404](#)
 - setting installer node [408](#)
 - upgrade [543](#), [552](#)

- spectrumscale installation toolkit (*continued*)
 - upgrade flow chart [545](#)
 - upgrade process flow [545](#)
 - upgrading [543](#), [552](#)
- ssh command [233](#)
- start replication
 - AFM [90](#)
- starting GPFS [236](#)
- stat cache [14](#)
- stat() system call [14](#), [21](#)
- stop replication
 - AFM [90](#)
- Storage Area Network (SAN)
 - disk considerations [238](#)
- Storage configuration [310](#)
- storage disk subsystems [310](#)
- storage management
 - filesets [3](#)
 - policies [3](#)
 - storage pools [3](#)
- storage policies [34](#)
- storage policies for object [34](#)
- storage pools [3](#)
- storage system [310](#)
- stretch cluster [431](#)
- strict replication [262](#)
- structure of IBM Spectrum Scale [4](#)
- substitution variables [312](#)
- support
 - failover [2](#)
- support and limitation for Windows [458](#)
- supported NFS clients in IBM Spectrum Scale [303](#)
- SUSE Linux Enterprise Server and Linux RPMs (update), extracting [358](#)
- synchronous
 - operations [48](#)
- synchronous or asynchronous operations
 - AFM [48](#)
- system calls
 - open [19](#)
 - read [19](#)
 - stat() [21](#)
 - write [20](#)
- system health
 - events [140](#)
 - mmhealth command [140](#)
 - thresholdmonitoring [140](#)
- systemd
 - planning [289](#)

T

- Thanks [523](#)
- Thin provision
 - TRIM-supported NVMe SSDs [245](#)
- Thin provisioned disks)
 - creation of [240](#)
- Thin provisioning
 - features [240](#)
- tiebreaker disks [224](#)
- token management
 - description [9](#)
 - large clusters [2](#)
 - system calls [19](#)

- token management (*continued*)
 - use of [2](#)
- toolkit installation [386](#), [442](#)
- Tracefmt program, installing [466](#)
- Tracelog program, installing [466](#)
- tracing
 - mmprotocoltrace command [140](#)
- transparent cloud tiering
 - considerations [338](#)
 - creating a node class [475](#)
 - firewall recommendations [334](#)
 - hardware requirements [329](#)
 - IBM Cloud Object Storage considerations [541](#)
 - installing
 - on GPFS nodes [476](#)
 - interoperability and limitations [175](#)
 - overview [170](#)
 - planning [329](#), [335](#)
 - security [338](#)
 - software requirements [329](#)
 - tuning and configuring [332](#)
 - upgrade to 1.1.2 [531](#)
 - upgrade to 1.1.2.1 [532](#)
 - upgrade to 1.1.3 [532](#), [533](#)
 - upgrade to 1.1.4 [534](#), [535](#)
 - upgrade to 1.1.5 [536](#)
 - upgrade to 1.1.6 [537](#)
 - working mechanism [170](#)
- Transparent Cloud Tiering
 - backup considerations [340](#)
- tuning system for transparent cloud tiering [332](#)

U

- Ubuntu Linux packages (update), extracting [358](#)
- UID and GID requirements on the cache and home clusters [341](#)
- uncached
 - files [47](#)
- unified file and object access
 - authentication [325](#)
 - authentication planning [325](#)
 - deploying [449](#)
 - high-level workflow [449](#)
 - identity management modes [325](#), [326](#)
 - objectization schedule [325](#)
 - objectizer planning [325](#)
 - overview [34](#)
 - planning [325](#), [326](#)
 - prerequisites [326](#)
- unified file and object access modes
 - planning [325](#), [326](#)
- uninstall
 - GPFS permanently [495](#)
 - transparent cloud tiering [503](#)
- UNIX and Windows (mixed) cluster, configuring [470](#)
- UNIX, Identity Mapping for Unix (IDMU) / RFC 2307 Attributes [470](#)
- unplanned maintenance
 - IW cache [87](#)
- update PTF notification [209](#)
- update SUSE Linux Enterprise Server and Red hat Enterprise Linux RPMs, extracting [358](#)
- update Ubuntu Linux packages, extracting [358](#)

- upgrade
 - Cloud services [531](#)
- Upgrade
 - v4.2.0. to v4.2.1 or later [525](#)
 - v4.2.1. to v5.0.0 [524](#)
- upgrade considerations
 - performance monitoring [507](#)
 - protocols [507](#)
- upgrade from 1.1.1 to 1.1.2
 - transparent cloud tiering [531](#)
- upgrade rerun
 - installation toolkit [560](#)
- upgrade to Red Hat Enterprise Linux 8.x [585](#)
- Upgrades [306](#)
- upgrading
 - completing the upgrade [572](#)
 - GPUDirect Storage [517](#)
 - offline upgrade [506](#)
 - online upgrade [506](#), [507](#)
 - performance monitoring [507](#)
 - protocols [507](#)
 - supported paths [506](#)
 - toolkit [543](#), [552](#)
- upgrading from version 3.5 [590](#)
- upgrading IBM Spectrum Scale GUI [529](#)
- upgrading operating system [580](#), [583](#)
- upgrading OS [580](#), [583](#)
- user-defined node class
 - Cloud services [475](#)
- using CES [29](#)
- using mmbbackup [89](#)

V

- verifying
 - GPFS for AIX installation [455](#)
 - GPFS for Linux installation [374](#)
 - GPFS installation (Ubuntu) [373](#)
 - prerequisite software for AIX nodes [453](#)
 - prerequisite software for Linux nodes [354](#)
- verifying disk usage [266](#)
- verifying signature [357](#)
- viewing snapshots at home [79](#)

W

- WAN
 - considerations [331](#)
- WAN considerations [331](#)
- watch folder API [186](#), [542](#)
- Windows
 - access control on GPFS file systems [464](#)
 - allowing the GPFS administrative account to run as a service [471](#)
 - antivirus software [462](#)
 - assigning a static IP address [466](#)
 - case sensitivity [461](#)
 - configuring [466](#)
 - configuring a mixed Windows and UNIX cluster [470](#)
 - configuring the GPFS Administration service [472](#)
 - creating the GPFS administrative account [471](#)
 - differences between GPFS and NTFS [463](#)
 - disabling the firewall [466](#)

- Windows (*continued*)
 - drive letter [265](#)
 - file name considerations [461](#)
 - file system name considerations [460](#)
 - GPFS limitations [459](#)
 - Identity Mapping for Unix (IDMU) / RFC 2307 Attributes [470](#)
 - installation procedure [457](#)
 - installing and configuring OpenSSH [472](#)
 - installing Cygwin [466](#)
 - installing GPFS on Windows nodes [468](#), [469](#)
 - joining an Active Directory domain [466](#)
 - overview [457](#)
 - prerequisites [465](#)
 - static IP address, Windows [466](#)
 - support and limitations [458](#)
- Windows HPC server, configuring [474](#)
- worker1Threads on primary cluster [343](#)
- workerThreads on cache cluster [342](#)
- Working of
 - Cloud data sharing [171](#)
- workload profiling [310](#)
- Write Once Read Many solution [174](#)
- write operation
 - buffer available [21](#)
 - buffer not available [21](#)
 - token management [21](#)



Product Number: 5641-DM1
5641-DM3
5641-DM5
5641-DA1
5641-DA3
5641-DA5
5737-F34
5737-I39
5765-DME
5765-DAE

SC27-9866-01

