Redbooks
ibm.com/redbooks

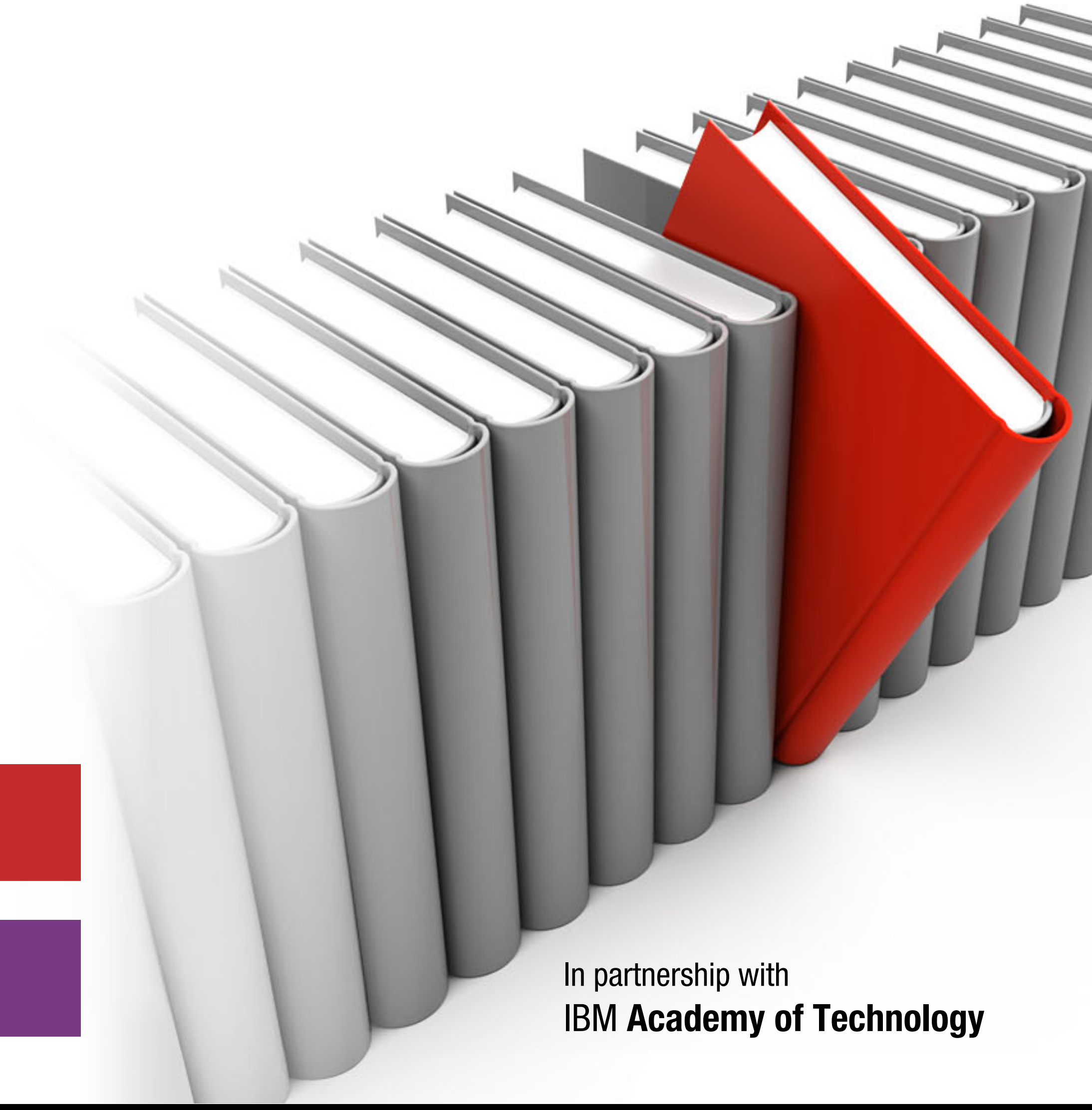
IBM Spectrum Scale Security

Felipe Knop

Sandeep R. Patil

Alifiya Kantawala

Larry Coyne



Cloud

Storage

In partnership with
IBM **Academy of Technology**

IBM

Redpaper

International Technical Support Organization

IBM Spectrum Scale Security

September 2018

REDP-5426-01

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

Second Edition (September 2018)

This edition applies to Version 5, Release 0, Modification 1 of IBM Spectrum Scale (product number 5725-Q01).

Contents

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

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.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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.

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.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, 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 http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

developerWorks®
GPFS™
Guardium®
IBM®
IBM Cloud™
IBM Elastic Storage™
IBM Spectrum™
IBM Spectrum Archive™
IBM Spectrum Control™
IBM Spectrum Protect™
IBM Spectrum Scale™
Redbooks®
Redpaper™
Redbooks (logo) ®
Storwize®
Tivoli®
WebSphere®

The following terms are trademarks of other companies:

SoftLayer, are trademarks or registered trademarks of SoftLayer, Inc., an IBM Company.

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Preface

Storage systems must provide reliable and convenient data access to all authorized users while simultaneously preventing threats coming from outside or even inside the enterprise.

Security threats come in many forms, including unauthorized access to data, data tampering, denial of service, and obtaining privileged access to systems.

According to the Storage Network Industry Association (SNIA), data security in the context of storage systems is responsible for safeguarding the data against theft, prevention of unauthorized disclosure of data, prevention of data tampering, and accidental corruption. This process ensures accountability, authenticity, business continuity, and regulatory compliance.

Security for storage systems can be classified as follows:

- Data storage (data at rest, which includes data durability and immutability)
- Access to data
- Movement of data (data in flight)
- Management of data

IBM® Spectrum Scale is a software-defined storage system for high performance, large-scale workloads on-premises or in the cloud.

IBM Spectrum™ Scale addresses all four aspects of security by securing data at rest (protecting data at rest with snapshots, backups, and immutability features) and securing data in flight (providing secure management of data, and secure access to data by using authentication and authorization across multiple supported access protocols). These protocols include POSIX, NFS, SMB, Hadoop, and Object (REST). For automated data management, it is equipped with powerful information lifecycle management (ILM) tools that can help administer unstructured data by providing the correct security for the correct data.

This IBM Redpaper™ publication details the various aspects of security in IBM Spectrum Scale™, including the following items:

- Security of data in transit
- Security of data at rest
- Authentication
- Authorization
- Hadoop security
- Immutability
- Secure administration
- Audit logging
- Security for transparent cloud tiering (TCT)
- Security for OpenStack drivers

Unless stated otherwise, the functions that are mentioned in this paper are available in IBM Spectrum Scale.

Authors

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Poughkeepsie Center.

Felipe Knop is a Senior Technical Staff Member working in software development in the IBM Spectrum Scale team, where he is the File System scrum core architect and also focuses on cluster management and file encryption. In addition, he was the security architect and the technical release lead. He has significant experience in the architecture and development of complex distributed subsystems in IBM Spectrum Scale and previously in Reliable Scalable Cluster Technology (RSCT). He holds a Ph.D. degree in computer science from Purdue University.

Sandeep R. Patil is a Senior Technical Staff Member who works as a Storage Architect with IBM System Labs. He has over 18 years of product architecture and design experience. Sandeep is an IBM Master Inventor, an IBM developerWorks® Master Author, and a member of the IBM Academy of Technology. Sandeep holds a Bachelor of Engineering (Computer Science) degree from the University of Pune, India. He is recognized and listed by Wikipedia in the World Wide Prolific Inventors list.

Alifiya Kantawala is the Information Developer Lead with the IBM India Systems Development Lab (ISDL) ID team. She is responsible for leading the ID team on the customer-facing documentation for various storage products, such as IBM Spectrum Scale, IBM Elastic Storage™ Server, and IBM Storwize® V7000 Unified. She has over 17 years of experience in product and training documentation. She holds a Bachelor’s degree in Commerce and Professional Diploma in Software Technology and System Management.

Larry Coyne is a Project Leader at the International Technical Support Organization, Tucson Arizona center. He has 35 years of IBM experience with 23 in IBM storage software management. He holds degrees in Software Engineering from the University of Texas at El Paso and Project Management from George Washington University. His areas of expertise include client relationship management, quality assurance, development management, and support management for IBM storage software.

Thanks to the following people for their contributions to this project:

The authors want to acknowledge the valuable contributions from the following individuals:

Piyush Chaudhary, Cheng Ding, Dileep Dixith, Sasikanth Eda, Shuo Feng, Deepak Ghuge, Nils Haustein, Kaustubh Katruwar, Ingo Meents, Kumaran Rajaram, John Olson, Aaron Palazzolo, Christof Schmitt, Gil Sharon, Gaurang Tapase, Carl Zetie, Yong Zheng, Smita J Raut, Deepavali M Bhagwat, Subashini Balachandran, Andreas Koeninger, Jacob M Tick, Amey P Gokhale.

In addition, the authors want to acknowledge the authors of helpful “How To” documents that are published as “Techdocs”:

Sandeep Bazar, Mamdouh Khamis, Chetan Kulkarni, Bill Owen

The authors also appreciate the valuable feedback from reviewers of early drafts of this paper:

Mathias Bjoerkqvist, Willard Davis, Monty Poppe, Norbert Schuld, Kuei-Yu Wang-Knop, Malahal R Naineni, Norbert Schuld, Abhishek Dave, Asmahan A Ali, Alexander Wolf, Gautam Shah, Muthuannamalai Muthiah, Rajan Mishra, Ted Hoover, Christina Lara

Now you can become a published author, too

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time. Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks® publications in one of the following ways:

- Use the online **Contact us** review Redbooks form:

  ibm.com/redbooks

- Send your comments in an email:

  redbooks@us.ibm.com

- Mail your comments:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

Secure data in transit

Data in transit, also referred to as *data in motion* or *data in flight*, is data that is being accessed over a network, and can therefore be intercepted by malicious users in the network. Based on business needs or on the sensitivity of your data, you need to protect it by using encryption over the wire.

IBM Spectrum Scale is a state-of-the-art storage platform that is ideal for data lake use cases because data can be accessed by using different interfaces, including Object (S3 and SWIFT-based REST interfaces), NFS, SMB, Hadoop, and standard file system access by using the IBM Spectrum Scale Client. IBM Spectrum Scale can ensure that data in motion is secure relative to the access method. Authentication and authorization aspects are covered in Chapter 3, “Authentication” on page 15 and Chapter 4, “Authorizing protocol users” on page 21. In the Chapter 11, “Security for AFM” on page 71, security considerations for AFM are covered.

This chapter includes the following sections:

- Secure cluster communication
- Secure access of IBM Spectrum Scale file interfaces
- Secured object data access
- Secure access with IBM Spectrum Scale HDFS Transparency
- References

1.1 Secure cluster communication

IBM Spectrum Scale is a clustered file system where data can be accessed across the cluster as a single namespace. To ensure secure communication and data transfer across the cluster, IBM Spectrum Scale has the concept of *security modes*. The security mode of a cluster determines the level of security that the cluster provides for communications between nodes in the cluster, and for communications between clusters. The level of security depends on the configured modes:

- EMPTY

  The sending node and the receiving node do not authenticate each other, do not encrypt transmitted data, and do not check data integrity. This mode is not recommended.

- AUTHONLY

  The sending and receiving nodes authenticate each other, but they do not encrypt transmitted data and do not check data integrity. This mode is the default for new clusters in IBM Spectrum Scale V4.2 or later.

- Cipher

  The sending and receiving nodes authenticate each other, encrypt transmitted data, and check data integrity. To set this mode, you must specify the name of a supported cipher, such as AES128-GCM-SHA256.

1.1.1 Important commands for secure cluster communication

A public/private key pair is generated by using the **mmauth** command:

```
mmauth genkey new
```

This command is used when either the AUTHONLY or Cipher mode is adopted. For remote cluster access, the system administrators of the two clusters are responsible for exchanging public keys.

You can display the security mode by using the **mmlsconfig** command:

```
mmlsconfig cipherlist
```

You can change the security mode by using the **mmchconfig** command:

```
mmchconfig cipherlist=security_mode
```

The set of supported ciphers is displayed by using the **mmauth** command:

```
mmauth show ciphers
```

When users are accessing a file system from another cluster, the cluster that owns a file system can designate a different security level for each connecting cluster. To specify a different security level for different clusters requesting access to a cluster, use the **mmauth -l cipherList** command. For more information, see IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_multsecl.htm

1.2 Secure access of IBM Spectrum Scale file interfaces

The following techniques can be used for secure access of IBM Spectrum Scale file interfaces:

- Secure access with SMB
- Secured NFS transfers

1.2.1 Secure access with SMB

The Server Message Block (SMB) protocol supports per-export encryption, which allows you to selectively enable or disable encryption on a per-SMB export basis. You can have some exports hosting sensitive data that is enabled for encryption, and others that do not host sensitive data, operate in non-encryption mode.

SMB protocol version 3 has capabilities to provide tighter security for the data transfers:

- Secured dialect negotiation
- Improved signing
- Secured transmission

Dialect negotiation is used to identify the highest-level dialect both the server and the client can support. The system administrator can enable SMB encryption by using the **`smb encrypt`** setting at the export level. The following three modes are available for secured SMB access:

- Automatic (auto)
- Mandatory
- Disabled

When the mode is set to `auto`, SMB encryption is offered, but not enforced. Mandatory means that SMB encryption is required and disabled means that SMB encryption cannot be negotiated. When the SMB services are enabled, SMB encryption is configured in the automatic (`auto`) mode by default.

You can either enable or disable encryption of the data in transit by using the **`mmsmb export create`** command, as shown in the following example:

```
# mmsmb export create secured_export  /ibm/gpfs0/secured_export --option "smb
encrypt=mandatory"
```

The `smb encrypt` option allows administrators to dictate whether the remote SMB client is allowed or required to use SMB encryption. For more information, see the **`mmsmb export`** command.

Note: Selecting encrypted traffic reduces throughput because smaller packet sizes must be used and the encryption of the data impacts the process.

1.2.2 Secured NFS transfers

The following security methods can be used with Network File System version 4 (NFSv4) protocol:

- Enabling squashing
- Using Kerberos
- Enabling port security

Enabling squashing

File requests that are made by the root user of the client system are considered a potential threat. By default, root user requests are treated as though they were made by the user `nobody` on the server (“root squash”). If you disable squashing, the root user of the client gets the same level of access to files on the system as the `root` user on the server.

The following command can be used to disable root squashing for an NFS export and set of clients:

```
mmnfs export change <nfs export path> --nfschange "<nfs
clients>(SQUASH=no_root_squash)"
```

For more information about the **`mmnfs`** command, see IBM Knowledge Center:

https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_mmnfs.htm

Using Kerberos

Kerberos is a network authentication protocol that can also ensure secure communication over a network. You can use Kerberos instead of local UNIX UIDs and GIDs to authenticate users. NFS with Kerberos can operate in three modes to provide improved security:

- Kerberos v5: Authentication (only) of the data traffic between the NFS client and server.
- Kerberos v5 with integrity: Authentication and data integrity of the data traffic between the NFS client and server.
- Kerberos v5 with privacy: Authentication and encryption of data traffic between the client and the server. This method is the most secure but can impact performance due to heavy processing.

For instructions about how to configure IBM Spectrum Scale to use NFS service over Kerberos, see the *Accessing IBM Spectrum Scale over NFS Kerberos using LDAP and MIT KDC* techdoc, at the following link:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD106395

Enabling port security

You can enable or disable port security for all communications between the NFS client and the NFS server. When port security is enabled, the system does not allow access to the requests that originate from ports where the port number is greater than the hardcoded threshold value of `1024`. TCP ports 1 - 1024 are reserved for use by root and are sometimes referred to as “secure ports.” A non-root user cannot bind to these ports. Adding the secure option prevents a malicious non-root user on the client from opening a spoofed NFS dialogue on a non-reserved port.

Note: These NFS security features can be configured per NFS export by the system administrator based on your requirements.

1.3 Secured object data access

IBM Spectrum Scale provides access to the Object Storage through Swift or the S3 REST API, and unified file and object access where objects can be accessed as files from file protocols and vice versa. For secure communication for IBM Spectrum Scale Object Storage and the clients, the system administrator must configure a load balancer, for example, *HAProxy*.

For SSL termination, traffic encryption, and load balancing of the requests, the load balancer must be set up on an external system that is not a part of the IBM Spectrum Scale cluster. For more information about how to use HAProxy for load balancing Object requests, see the following documentation:

http://www.haproxy.org/#docs

An SSL termination proxy is a proxy server that is used to receive incoming SSL/TLS connections. It then decrypts the SSL/TLS and passes on the decrypted request to the object server proxy. This system assumes that the network between the SSL termination proxy and the object server is secure.

For more information about how to configure IBM Spectrum Scale Swift and Keystone with HAProxy, see *Configuring IBM Spectrum Scale Swift and Keystone with HAProxy*, found at:

https://mkguru.wordpress.com/2015/08/12/configuring-spectrum-scale-swift-and-keystone-with-haproxy

1.4 Secure access with IBM Spectrum Scale HDFS Transparency

IBM Spectrum Scale Hadoop Distributed File System (HDFS) transparency offers a set of interfaces that allow applications to use HDFS Clients to access IBM Spectrum Scale through HDFS Remote Procedure Call (RPC) requests. When you are using the open source HDFS, all data transmissions and metadata operations are done through RPCs and processed by NameNode and DataNode services within HDFS.

The IBM Spectrum Scale HDFS transparency implementation integrates both NameNode and DataNode services, and responds to the request as HDFS would. HDFS transparency can be configured in “Kerberos Mode”, encrypting and securing data transfers and RPCs from the clients to the NameNode and DataNode.

For more information, see Chapter 8, “Hadoop security” on page 55.

1.5 References

The following websites are useful for further research:

- Security modes:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_securitymode.htm

- Accessing a remote IBM Spectrum Scale file system:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_admmcch.htm

- Using multiple security levels for remote access:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_multsecl.htm
- Securing NFS data transfer:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_configprotdatasec.htm
- *Accessing IBM Spectrum Scale over NFS Kerberos using LDAP and MIT KDC*:

  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD106395
- *Configuring IBM Spectrum Scale Swift and Keystone with HAProxy*:

  https://mkguru.wordpress.com/2015/08/12/configuring-spectrum-scale-swift-and-keystone-with-haproxy
- Command reference (**mmnfs** command):

  https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_mmnfs.htm

2

Secure data at rest

IBM Spectrum Scale provides secure data at rest by using file-level encryption. User files are encrypted on disk and are not retrievable unless the correct encryption keys are available. That configuration ensures that the privacy of the data on the disks is maintained if physical access to the storage becomes compromised.

Encryption is integrated with the IBM Spectrum Scale policy engine. You can create rules to determine which files are encrypted and how. Encryption keys are maintained at a separate key manager outside the file system, and possibly outside the cluster. Encrypted files can be securely erased because after a key that is used to access a file is deleted, the content of that file is no longer retrievable.

Encryption is available with IBM Spectrum Scale Advanced Edition and Data Management Edition. For more information about encryption, see IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_encryption.htm

This chapter includes the following sections:

- Encryption keys
- Encryption policies
- The remote key management service
- Illustration of the encryption policy
- Highly available key servers
- Multicluster and disaster recovery
- Secure deletion
- NIST and FIPS
- Encryption: Performance impact
- References

2.1 Encryption keys

To allow flexibility when changing the set of encryption keys that are used for encrypting a file, including the secure erase capability, a two-level encryption key infrastructure is in place. First, each individual file is encrypted with a randomly generated FEK, which is chosen at file-creation time. Then, the FEK is encrypted ("wrapped") with the master encryption key (MEK), which is the key that is retrieved from the external key server. Although each file has its individual FEK, multiple files (even all the files in a file system) can share an MEK.

The mechanism that is used to "wrap" FEKs is flexible and enables "and" or "or" semantics when multiple MEKs are specified for a file. For example, when two MEKs are used, a file can be opened only if both MEKs M1 and M2 can be retrieved from the key server. You can also allow the file to be opened if either MEK M1 or M2 is present. The behavior is determined by how you set up the encryption policy rules. The simplest case is one where all the files use a single MEK to wrap each file's FEK.

2.2 Encryption policies

Encryption policy rules are used to determine which files are encrypted, with which algorithm, and using which MEKs. Encryption rules are configured with the **`mmchpolicy`** command. After the policies are in place, they apply to any newly created file, and the file's encryption attributes are set. After the file is created, it is no longer possible to alter its status. That is, if it is an encrypted file, it is no longer possible to turn it into a non-encrypted file, and vice versa. However, it is possible to "rewrap" the file by using a migration policy so that a file's FEK is rewrapped with new MEKs.

The encryption rules determine which files are to be encrypted and what MEKs to use. For example, the file selection can be made based on these parameters:

- Encrypt all files in the file system in the same way.
- Select files in a particular fileset.
- The name of the file (for example, encrypt all files whose name ends with the `.enc` extension).
- User ID or Group ID.

Two encryption rules determine how files are encrypted. The **`ENCRYPTION IS`** rule specifies how a file is encrypted, including the encryption algorithm and key length, and how to combine and wrap keys. Finally, this rule specifies what MEKs to use. The MEKs are described by a `KeyId:RkmId` pair, where the first component is the identifier of the key in its key server, and the second component represents the identity of the key server.

The **`SET ENCRYPTION`** rule is used to select the files to which the **`ENCRYPTION IS`** rule applies.

Example 2-1 shows a sample of encryption policy rules.

Example 2-1 Example encryption policy rules

```
RULE 'myEncRule1' ENCRYPTION 'E1' IS
         ALGO 'DEFAULTNISTSP800131A'
         KEYS('1:RKM_1', '2:RKM_2')
RULE 'Encrypt files with extension enc1 with rule E1'
         SET ENCRYPTION 'E1'
         FOR FILESET('fs1')
         WHERE NAME LIKE '%.enc1'
```

Files in fileset `fs1` and that have an extension that is equal to `.enc1` are encrypted with a 256-bit FEK by using the AES block cipher in XTS mode. The FEK is preprocessed with a hash-based message authentication code (HMAC) with SHA-512, and the FEK is then wrapped with an AES key wrap, with keys `1:RKM_1` and `2:RKM_2` combined during one round of XOR followed by one round of HMAC with SHA-512. `'DEFAULTNISTSP800131A'` is a shortcut for “256-bit FEK, AES in XTS mode, and preprocessing the FEK by using HMAC with SHA-512.”

Because existing unencrypted files cannot be encrypted, to encrypt existing files you must copy them into new files after an appropriate encryption policy is put in place.

2.3 The remote key management service

MEKs are retrieved from an external remote key management (RKM) service. Key management is provided by the IBM Security Key Lifecycle Manager software, which is licensed separately. Alternatively, starting with IBM Spectrum Scale V4.2.1, you can use the Vormetric DSM key server. Both key servers implement the Key Management Interoperability Protocol (KMIP) standard, and that protocol is used by the nodes in the cluster to retrieve keys.

Although the key server does not need to be on one of the nodes in the IBM Spectrum Scale cluster, it must be network-reachable from each node in the cluster to allow the keys to be retrieved.

After the key management service is installed, it must be integrated with IBM Spectrum Scale to authorize nodes in the cluster to retrieve the MEKs. At a high level, the following steps are required:

1. Create a server certificate and export it.
2. Create a *device group* (IBM Security Key Lifecycle Manager) and keys for the device group.
3. Create a client keystore, where a private key and a client certificate are stored, and also store the trusted IBM Security Key Lifecycle Manager server certificate in it.
4. Create the `/var/mmfs/etc/RKM.conf` file, with stanzas representing the RKM.
5. Mark the client certificate as trusted in the device group (server side).

Each stanza in the RKM.conf file contains the stanza name (identification of the RKM), and this information:

- Location of the key server (Uniform Resource Identifier (URI)), including host name and port number
- Location of keystore and its passphrase
- Client certificate label
- Device group (tenant name)

After the RKM configuration is completed, including the creation of keys, encryption policy rules can be created to determine how the files are encrypted. To install the encryption policies for the file system, use the **mmchpolicy** command.

With Version 4.2.1, an improved mechanism was introduced to configure IBM Security Key Lifecycle Manager key servers, retrieve server certificates, and create device groups (tenants) and keys, and stanzas in the RKM.conf file. The mechanism includes propagation of the needed configuration across the cluster. This new feature is described in IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_encryption.htm

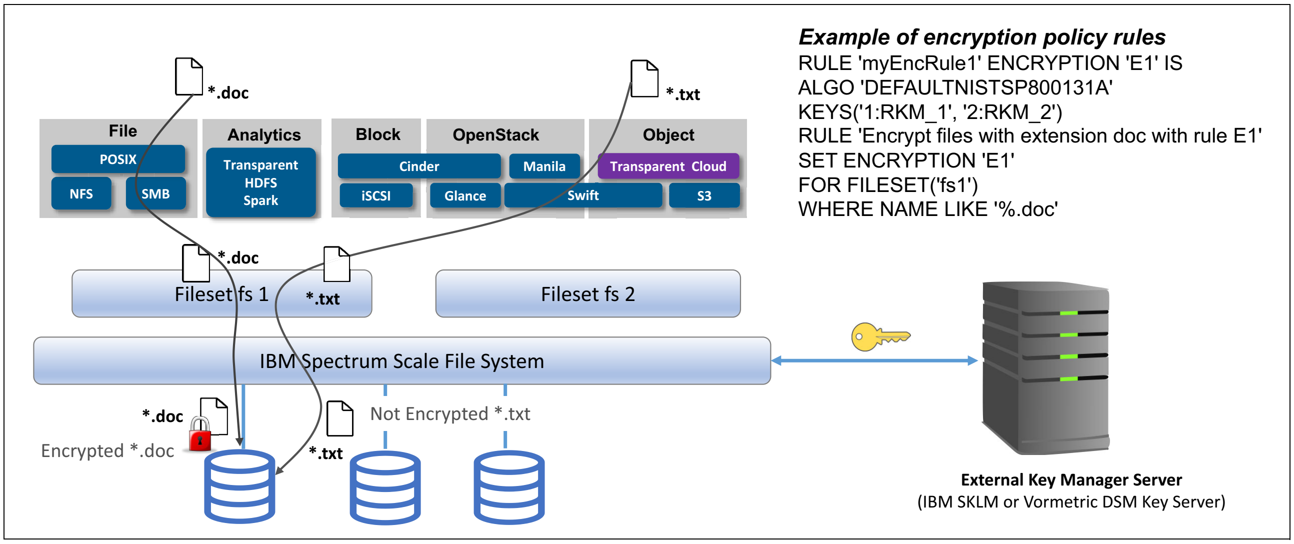
The original setup is described in IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_encryptionenv_regular.htm

2.4 Illustration of the encryption policy

Figure 2-1 on page 11 illustrates a file encryption example using encryption policy rules. Only the files located in fileset fs 1 and having an extension equal to .doc are encrypted with a 256-bit file encryption key (FEK) by using the AES block cipher in XTS mode.

The FEK is preprocessed with a hash-based message authentication code (HMAC) with SHA-512, and then it is wrapped with an AES key wrap, with keys 1:RKM_1 and 2:RKM_2 combined during one round of XOR followed by one round of HMAC with SHA-512. 'DEFAULTNISTSP800131A' is a shortcut for “256-bit FEK, AES in XTS mode, and preprocessing the FEK by using HMAC with SHA-512.”



Example of encryption policy rules

```
RULE 'myEncRule1' ENCRYPTION 'E1' IS
ALGO 'DEFAULTNISTSP800131A'
KEYS('1:RKM_1', '2:RKM_2')
RULE 'Encrypt files with extension doc with rule E1'
SET ENCRYPTION 'E1'
FOR FILESET('fs1')
WHERE NAME LIKE '%.doc'
```

Figure 2-1 Example of encryption policy rules

2.5 Highly available key servers

Ensuring that keys remain available is crucial to continuous file system operations. Both IBM Security Key Lifecycle Manager and Vormetric DSM have a high availability feature that allows key servers to be deployed in a high-availability configuration. In this configuration, the same keys can be retrieved from “cloned” servers. This capability can be used by specifying multiple URIs in the RKM stanza in the `RKM.conf` file.

After multiple key servers are specified for an RKM, IBM Spectrum Scale attempts to contact each one in succession until the key can be retrieved.

Cloned servers can be used to improve the performance of key retrieval. By controlling the order of the cloned servers for an RKM in the `RKM.conf` file across the multiple nodes, you can achieve load balancing on the access to the servers, or ensure that the server that is “closer” to the node is attempted first.

For a Vormetric DSM implementation with HA capability contact Vormetric support for the current code level requirements.

2.6 Multicluster and disaster recovery

If an encrypted file system is made available through remote mounts, then the remote cluster nodes require network reachability to the key server. In some deployments, the key servers might be at the home cluster, but you can locate key servers in a high-availability configuration on both home and remote clusters.

Although setting up multiple key servers in a high-availability configuration is important to ensure that the MEKs remain available, it is especially important in a disaster recovery environment. Place at least one key server on each site to ensure that keys remain available if access to an entire site is lost.

2.7 Secure deletion

Secure deletion of a set of files can be achieved by a combination of a normal file deletion and removal of the MEKs that are used to encrypt those files. Complete the following steps:

1. Remove the files by using the **rm** command (or the **unlink()** system call).
2. Create one or more new MEKs in the key server.
3. Modify the encryption policies so that the instances of MEKs that are used by those deleted files (and shared with other files in the file system) are replaced by the newly created keys.
4. Create and apply a migration (rewrapping) policy (use **CHANGE ENCRYPTION KEYS**) to scan all files, unwrap the wrapped FEK entries of files that are wrapped with the old key, and rewrap them with the new key. This step ensures that the FEKs of existing files are accessible in the future.
5. Remove the old key from the key server, which commits the secure deletion of all files that were previously unlinked.

2.8 NIST and FIPS

The encryption algorithms that are used for file encryption are all compliant with NIST Special Publication 800-131A. To ensure that secure communication between each node and the key server works correctly, the administrator must adjust the TLS settings on the key server to match the IBM GPFS™ **nistCompliance** configuration variable.

Because most clusters should be configured with **nistCompliance** set to SP800-131A (which is the default for new clusters created starting with V4.1), the transport mechanism for the key server should be set for NIST-800-131A compliance. For more information, see IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_encryptionenv_maintopic.htm

The **FIPS1402mode** configuration variable controls whether the use of security mechanisms that are based on cryptography is provided by software modules that are certified to conform with FIPS 140-2. If the variable (default value no) is set to yes, then encryption services are provided by the IBM Global Security Toolkit (GSKit), which is included with IBM Spectrum Scale. In some instances, Direct I/O operations are implemented completely in the kernel, including data encryption/decryption.

In those cases, cryptographic functions are provided by the Linux kernel. However, these are not used unless the **FIPS1402mode** variable is set to no or if the Linux kernel itself is compiled and configured in FIPS mode. That restriction ensures that only FIPS-certified mechanisms are used if the cluster is configured in FIPS mode (**FIPS1402mode** set to yes). When the kernel cryptographic functions cannot be used, encryption is performed in the user space with GSKit.

2.9 Encryption: Performance impact

Data in the `pagepool` remains in cleartext (unencrypted) and gets encrypted on its way to a local disk or to the NSD server. Similarly, encrypted data on disk is decrypted just after being retrieved from the disk, if the disk is local to the node.

For NSDs, the data is decrypted at the NSD client. Although encryption and decryption require extra processing, the impact is usually modest (less than 5%) for sequential or mixed workloads because most of the encryption activity occurs outside the application critical path, such as during either data prefetch or write-behind. The relative impact is reduced if the aggregate disk bandwidth is fully used. The relative impact can be higher if the disk access time is low.

To reduce the performance impact of encryption, hardware acceleration is automatically used in the following cases:

- Intel x86: AESNI instruction
- Power 8 in-core AES acceleration (which as of Version 4.2.1 is used both in non-FIPS and FIPS modes)

2.10 References

The following websites are useful for further research:

- Encryption chapter in the IBM Spectrum Scale documentation:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_encryption.htm

- Configuring encryption with IBM Security Key Lifecycle Manager: Regular setup:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_encryptionenv_regular.htm

3

Authentication

IBM Spectrum Scale, starting with Version 4.1.1, has integrated NFS, SMB, and Object support, with the object interface based on OpenStack Swift. The highly available functions are enabled by configuring Cluster Export Services (CES) on a subset of nodes of the IBM Spectrum Scale cluster. Such a subset of nodes is designated as protocol nodes, or CES nodes, which form a CES cluster.

This chapter includes the following sections:

- File interface
- Object interface
- References

3.1 File interface

SMB and NFS require user authentication. IBM Spectrum Scale supports various authentication directory servers:

- RFC2307 schema-compliant Lightweight Directory Access Protocol (LDAP) server
- Microsoft Active Directory (AD) server
- Network Information Service (NIS) server

File protocols can be configured with any one of the authentication sources at a point in time. In addition, you can configure authentication in your preferred way, which is referred to as the *user defined* authentication scheme. The supported authentication scheme matrix can be found in IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1ins_authconcept.htm

IBM Spectrum Scale identifies users and groups by numeric UID and GID. SMB clients use Security Identifiers (SIDs) when accessing shares, so if you want to have concurrent NFS and SMB access to IBM Spectrum Scale, you must provide a mapping between the SID and a valid UID with an associated default GID. Similarly, Kerberos principals that are presented during SMB and NFS access must be mapped into a valid UID and GID. This mechanism of Name-to-ID mapping is referred to as *ID mapping* in IBM Spectrum Scale terminology.

The concept of ID mapping appears in two different contexts in IBM Spectrum Scale, when mapping an SID to a GID for SMB access and when mapping UIDs between clusters that are connected by using multi-cluster. For a user to gain access to data over file protocols, both authentication and ID mapping must be successful.

Figure 3-1 demonstrates a high-level process flow of authentication and ID mapping for NFS and SMB with IBM Spectrum Scale protocols.

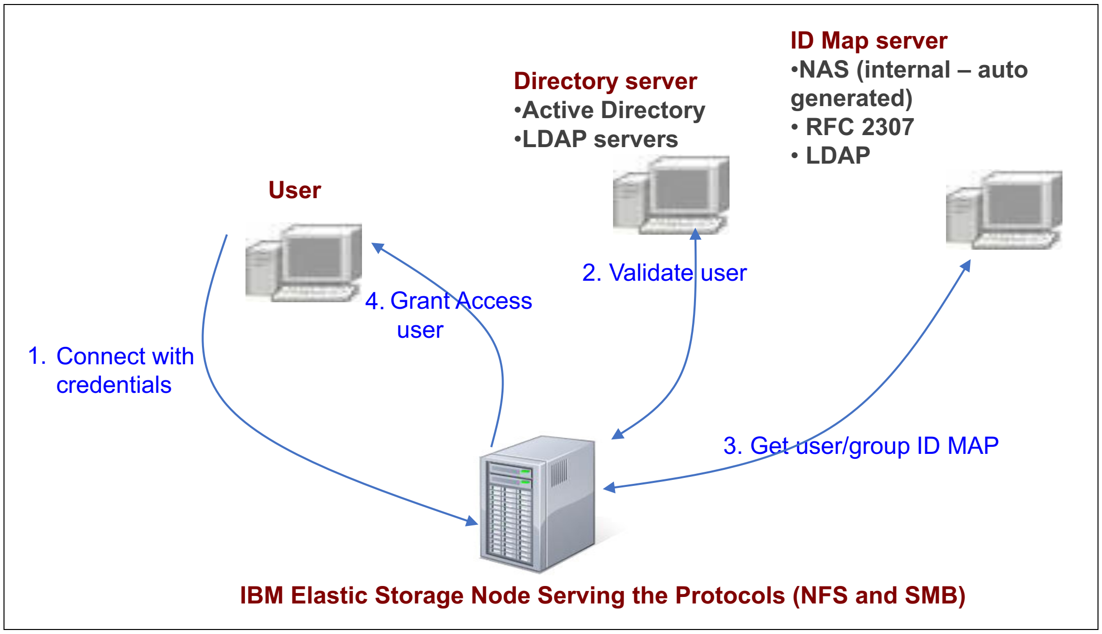


Figure 3-1 IBM Spectrum Scale Protocol Serving Protocols (NFS and SMB)

POSIX Interface

When using the POSIX interface via IBM Spectrum Scale client to access the data on an IBM Spectrum Scale file system, the authentication and ID mapping of the users/groups accessing the file system is dependent and leveraged from the user authentication and ID mapping done by the client operating system on which the clustered file system has been mounted.

The client operating system needs to be configured with the appropriate directory services (AD/LDAP/NIS) so that the users/groups are able to get the required UID/GID when logging in. These user credentials are used for seeking authorization when a file is accessed on the clustered file system through IBM Spectrum Scale clients.

3.1.1 Integration with an RFC2307 schema-compliant LDAP server

IBM Spectrum Scale supports LDAP servers that have RFC2307 schema-compliant entries for users, groups, and netgroups (for example, OpenLDAP and IBM Tivoli® Directory Server). To help support NTLM-based SMB access, the Samba schema must be imported on the LDAP server, and relevant Samba schema classes must be implemented by the user and group entries.

To achieve secure communication between IBM Spectrum Scale protocol nodes and the LDAP server, it is a good idea to choose the LDAP over the TLS-based authentication scheme. This authentication scheme ensures that authenticating a user against the claimed LDAP user entry and relevant information lookup on the LDAP server is secured.

3.1.2 Integration with a Microsoft Active Directory server

An AD server can be implemented in two different SID to GID mapping modes: auto or predefined. The auto generation of SID to UID mapping works when you have a single IBM Spectrum Scale cluster or all SMB data is accessed only through SMB. It is a good idea to implement predefined SID to UID mapping.

Most often, this is accomplished by storing UID and GID information in an AD server by using the RFC2307 attributes or RFC2307-compliant defined entries on a stand-alone LDAP server. The source of UID and GID information can be defined for each AD domain (native or in trust) while configuring file protocols for AD authentication.

3.1.3 Integration with a Network Information Service server

Integration with an NIS infrastructure server can be achieved in the NIS authentication scheme. When integrating with NIS, only the NFS service is allowed to be enabled for data serving.

3.1.4 Kerberos authentication

Kerberos authentication is supported by the AD and LDAP authentication schemes. When IBM Spectrum Scale file protocols are configured with AD, Kerberos-based authentication for SMB is automatically enabled. Kerberos-based access for NFS service requires extra configuration. To perform Kerberos-based access, you must access the CES cluster with the `netbios` name.

The netbios name is configurable and set while authentication scheme configuration is performed. Kerberos-based access over SMB and NFS protocols can be used with LDAP as well. To set up Kerberos with LDAP, file protocols must be configured with MIT Kerberos server.

3.1.5 Netgroups

Access to NFS exports is typically restricted by listing an allowed set of hosts in the export definition. NFS-based access can be secured by using netgroups instead of defining a list of hosts for each export. Securing NFS exports by using netgroup definitions that are stored on authentication servers is supported by LDAP and NIS-based authentication schemes.

3.2 Object interface

Similar to file interface protocols, an object interface access request on IBM Spectrum Scale is authenticated before serving the data. The IBM Spectrum Scale object store relies on the Keystone service to validate an incoming user before processing the object access request. Keystone is the identity validation service that is used by OpenStack services.

Figure 3-2 shows a high-level flow of IBM Spectrum Scale object interface access.

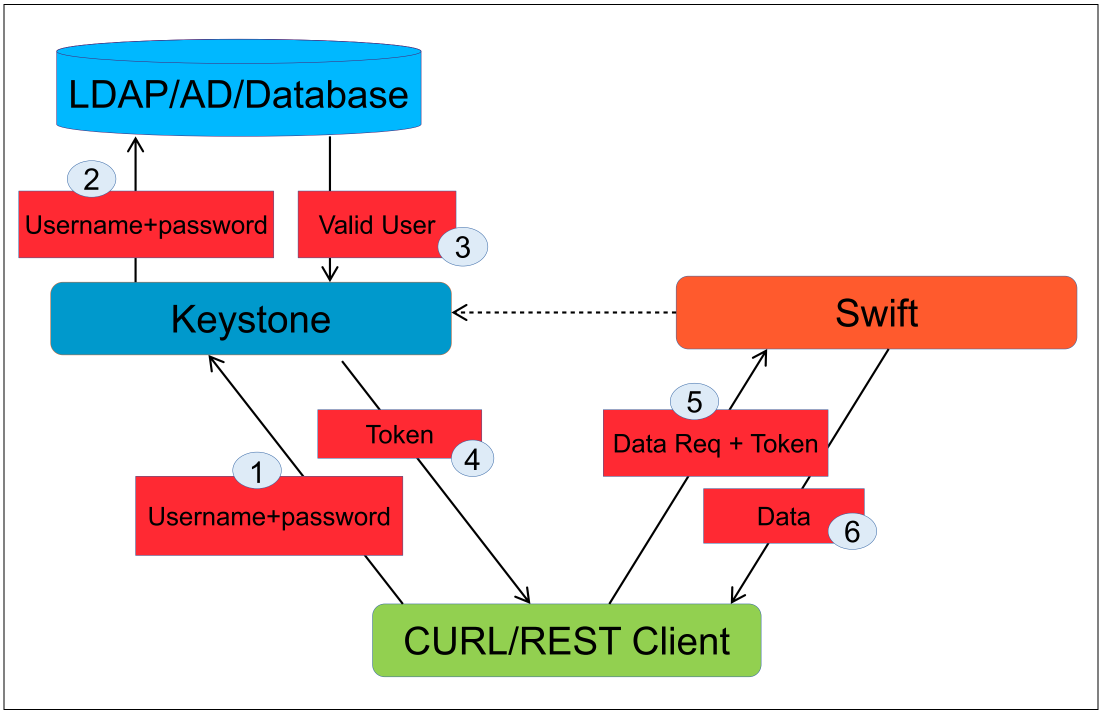


Figure 3-2 Basic object access request flow

IBM Spectrum Scale supports configuring Keystone with the following identity back ends:

- Microsoft AD server
- LDAP server
- Postgres database

The Identity back end is the source of the user name and password. Before sending a request to the IBM Spectrum Scale object store, the client sends a request to the Keystone service to obtain the token that is required for accessing the object service. The request to the Keystone service contains the user name and password. The Keystone service validates the user name and password with the configured Identity back end.

On successful validation of user name and password, Keystone returns a token to the user/client. The obtained token is used for object store access request. The token expires after the preset time. The token expiration period is configurable.

The allocated token can be revoked by the Keystone service by sending a token DELETE request. An object access request with a revoked token is rejected by the IBM Spectrum Scale object store.

Figure 3-3 depicts possible configurations in which an object interface identity service Keystone can be configured. It also reflects security aspects that are involved in communication across various entities.

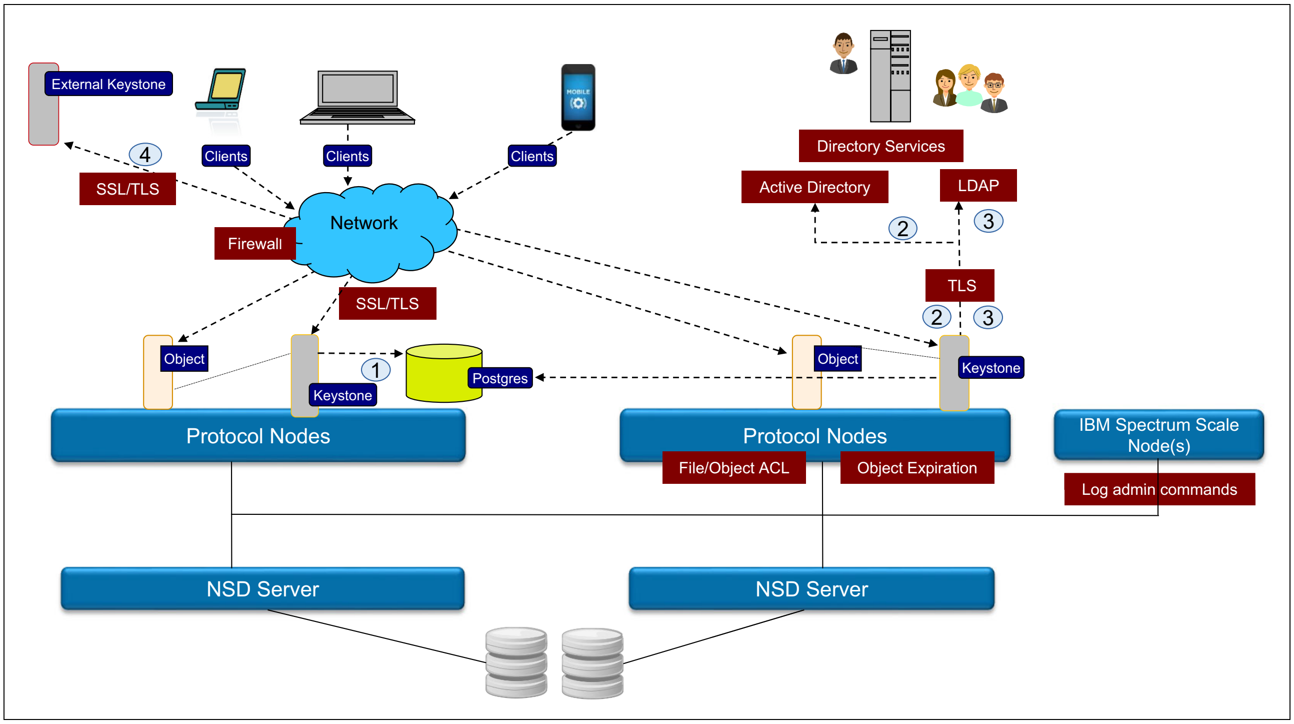


Figure 3-3 Object interface authentication configuration and secure communication aspects that are addressed by IBM Spectrum Scale

The IBM Spectrum Scale object interface can be configured for one of the following authentication schemes:

- Local

  The Keystone service refers to identities that are defined on the IBM Spectrum Scale CES cluster. User identities are stored locally in a Postgres database.

- Integration with AD server

  The Keystone service uses AD server as the identity back end.

- Integration with LDAP server

  The Keystone service uses LDAP server as the identity back end.

- Userdefined

  Integrating IBM Spectrum Scale with a Keystone service that is hosted by the user. In this case, the Keystone service is hosted by your infrastructure, and the management of the identity server is up to you.

In the first three types of object interface authentication schemes, the Keystone service runs on all protocol nodes for high availability.

3.2.1 Secure communication

Consider these items about secure communication:

- The Keystone service can be configured with `https` (SSL) for better security. The communication between the object/Keystone client and the Keystone service that is hosted on IBM Spectrum Scale is over `https` (SSL).
- IBM Spectrum Scale provides an option to configure the communication between the Keystone service and identity back end (Microsoft AD/LDAP) to be over TLS.
- To use features of the Keystone service that are not provided by the Keystone service that is hosted on IBM Spectrum Scale, configure an externally hosted Keystone service by using the `userdefined` object interface authentication scheme. The communication between the IBM Spectrum Scale object store and the external Keystone service can be configured to be established over `https` (SSL).

3.3 References

For more information, see IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1ins_authconcept.htm

4

Authorizing protocol users

Authorization means granting or denying access to resources. Authorization is applicable to an already authenticated identity. IBM Spectrum Scale supports access control lists (ACLs) to provide fine-grained access control to files and directories.

This chapter includes the following sections:

- Authorizing NFS and SMB users
- Authorizing Object (OpenStack Swift and S3) users
- References

4.1 Authorizing NFS and SMB users

IBM Spectrum Scale uses ACLs to authorize users who access the system through file protocols such as NFS and SMB. Although the file system supports two different types of ACLs (NFSv4 and POSIX), only NFSv4 ACLs are supported for the SMB protocol, and NFSv4 ACLs are preferred for general protocol usage because NFS protocol uses NFSv4 ACLs natively. In addition, the ACLs in the SMB protocol (also called *Security Descriptors*) can be mapped directly to and from the NFSv4 ACLs stored in the file system.

4.1.1 NFSv4 ACLs

Each file or directory in the file system grants permissions according to an ACL. Each ACL consists of one or more access control entries (ACEs). Each ACE allows or denies certain permissions to a principal (the user or group that is specified in the ACE). When the file system must determine whether a user is granted a requested set of permissions, the ACEs are evaluated in the order in which they are specified in the ACL. NFSv4 ACLs default to `deny`. If a requested permission is not explicitly granted in any ACE, then the permission request is denied.

Note: The NFSv4 standard also specifies `audit` and `alarm` ACEs that can be used for auditing access. These ACEs are not supported by IBM Spectrum Scale.

4.1.2 Inheritance in NFSv4 ACLs

Each entry in an NFSv4 ACL can be inherited. Inheritance applies only when new files or subdirectories are created in a directory. It functions like a template for new files and directories. All ACE entries from the directory that are marked as `inherit to file` or `inherit to directory` are put in the ACL of a newly created file or directory. The ACL entries in the directory can also be marked as `inherit only`, meaning that these entries are inherited according to the inheritance flag, but they are not used for permission checks on the parent directory.

When an SMB client system requests to change the ACLs of a directory tree (for example, from the Security window on a Microsoft Windows system), the client must traverse the tree and change the ACL. Depending on the number of files and directories in that tree, this operation might take a long time.

When a file or directory is created in IBM Spectrum Scale where the parent directory does not have inheriting ACL entries, default permissions are set on the new file or directory. The default permissions are created according to the modebits from the call that creates the file or directory, and the active `umask`.

4.1.3 ACLs in the SMB protocol

Client systems that access IBM Spectrum Scale through the SMB protocol can also set and query ACLs. These SMB ACLs are also referred to as *Windows ACLs* or *Security Descriptors*. The SMB ACL consists of two parts:

- A discretionary access control list (DACL), which consists of `allow` and `deny` entries that grant permissions.
- A system access control list (SACL), which is used to audit access, similar to the NFSv4 `alarm` and `audit` ACEs.

Because IBM Spectrum Scale does not support NFSv4 `alarm` and `audit` ACEs, it also does not support SACLs for SMB clients.

Principals in SMB ACL entries are security identifiers (SIDs). When accessing SMB exports in IBM Spectrum Scale, the SMB service maps between the SMB ACEs and NFSv4 ACEs, and maps the SID principals to the corresponding users and groups.

This mapping requires authentication and ID mapping to be configured properly in the cluster. ID mapping in IBM Spectrum Scale requires the user and that user's primary group to have a valid ID mapping. Secondary groups are silently dropped if there is no valid ID mapping. Consider this important point when using `deny` ACL entries, because that can result in the `deny` entry for a secondary group getting ignored, therefore resulting in additional permissions for the user. When all secondary groups have a valid ID mapping, this is not an issue.

SMB client systems running the Microsoft Windows operating system require a "canonical" order in the ACE entries where non-inherited `deny` entries must be listed first in the ACL, then non-inherited `allow` entries, then inherited `deny` entries, and last inherited `allow` entries.

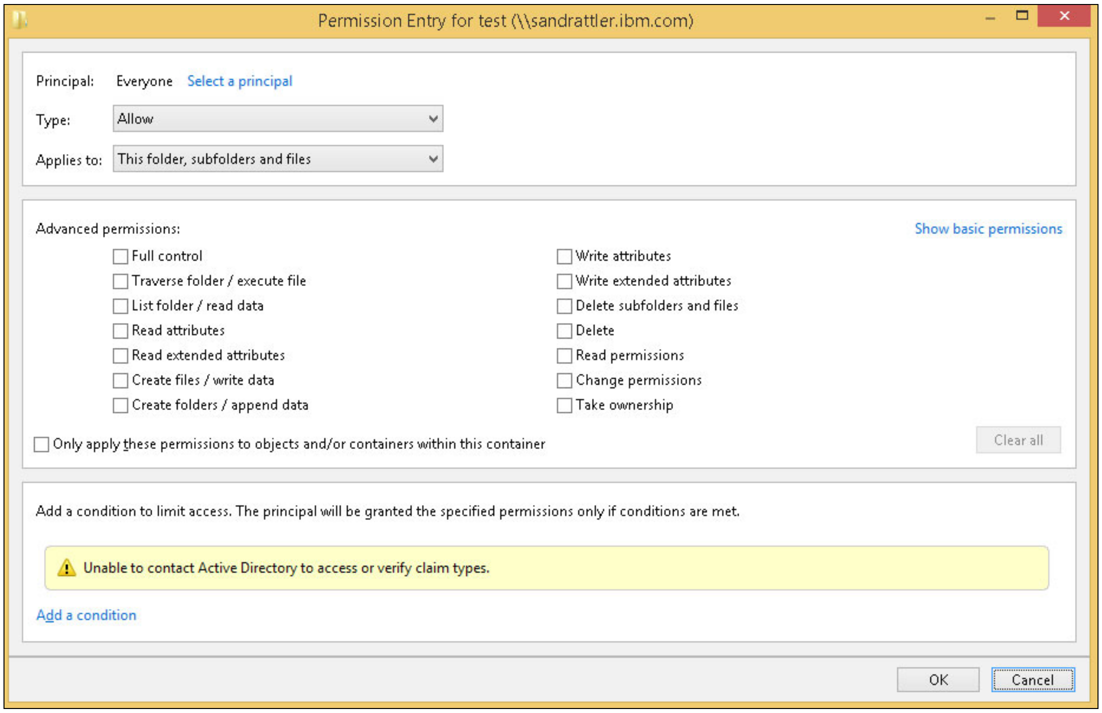
4.1.4 ACLs in the NFS protocol

Clients connecting through the NFSv4 protocol can query and set the NFSv4 ACLs that are stored in the file system directly. Clients connecting through the NFSv3 protocol can query only the subset of ACL entries that map to the POSIX modebits. In either case, access to files and directories is enforced on the IBM Spectrum Scale cluster according to the complete ACL.

4.1.5 Mapping between NFSv4 ACLs and SMB ACLs

IBM Spectrum Scale stores only one NFSv4 ACL per file or directory. When an SMB client requests the SMB ACL for a file or directory, the SMB service reads the NFSv4 ACL, maps it to an SMB ACL, and returns that to the SMB client. Similarly, when an SMB client writes an SMB ACL to a file or directory, the SMB service maps the SMB ACL to an NFSv4 ACL and stores that in the file system. The principals in the ACL entries are mapped according to the configured ID mapping in the IBM Spectrum Scale cluster.

The permissions in the SMB ACEs and NFSv4 ACEs can be mapped directly, as seen when comparing the available permission bits, which is shown in Figure 4-1.


Permission Entry for test (\\sandrattler.ibm.com)

Principal: Everyone Select a principal

Type: Allow

Applies to: This folder, subfolders and files

Advanced permissions: Show basic permissions

□ Full control
□ Traverse folder / execute file
□ List folder / read data
□ Read attributes
□ Read extended attributes
□ Create files / write data
□ Create folders / append data
□ Write attributes
□ Write extended attributes
□ Delete subfolders and files
□ Delete
□ Read permissions
□ Change permissions
□ Take ownership

□ Only apply these permissions to objects and/or containers within this container

Add a condition to limit access. The principal will be granted the specified permissions only if conditions are met.

Unable to contact Active Directory to access or verify claim types.

Add a condition

Figure 4-1 Permission Entry for test

The following code is the NFSv4 ACL for the same file.

```
special:everyone@:----:allow:FileInherit:DirInherit
 (-)READ/LIST (-)WRITE/CREATE (-)APPEND/MKDIR (-)SYNCHRONIZE (-)READ_ACL
(-)READ_ATTR  (-)READ_NAMED
 (-)DELETE    (-)DELETE_CHILD (-)CHOWN        (-)EXEC/SEARCH (-)WRITE_ACL
(-)WRITE_ATTR (-)WRITE_NAMED
```

Traverse folder / execute file in the SMB ACL maps to EXEC/SEARCH in the NFSv4 ACL, **List folder / read data** maps to READ/LIST, and so on. The following items are exceptions in this example:

- **Full Control** in the Windows Permission Entry window is not an actual permission bit, but only a shortcut to set all permission bits.
- The SYNCHRONIZE bit in the NFSv4 ACL maps to the SYNCHRONIZE bit in the SMB ACL. However, this is not an actual permission that is selected, and is not shown in the window.

The inheritance flags map between the Windows client, and the NFSv4 ACL: **Applies to: This folder, subfolder and files** maps to `FileInherit:DirInherit`.

Note: The `claim-based ACL` type is not supported in IBM Spectrum Scale.

4.1.6 Special ACL entries

In addition to referring to an explicit user or group, principals in NFSv4 ACL entries can also refer to special entries:

- The `special:owner@` principal grants permissions to the owner of the file. When mapping to an SMB ACL, a `special:owner@` entry that is flagged with `Inherit` is mapped to the CREATOR OWNER principal.
- The `special:group@` principal grants permissions to the owning group of the file. When mapping to an SMB ACL, a `special:group@` entry that is flagged with `Inherit` is mapped to the CREATOR GROUP principal.
- The `special:everyone@` principal is mapped to the `Everyone` principal in an SMB ACL.

4.1.7 SMB export ACLs

For SMB exports, an additional ACL can be defined for each SMB export. IBM Spectrum Scale allows the administration of SMB export ACLs through the **`mmsmb exportacl`** command. This is a second set of ACLs that is stored in the SMB server and does not map to the NFSv4 ACLs that are stored in the file system.

To have access to a file or directory through the SMB protocol, a user must have sufficient permissions in the SMB export ACL and in the file system ACL. The default of the SMB export ACL allows any access. With this default, only the file system ACLs are relevant for permission checking.

4.2 Authorizing Object (OpenStack Swift and S3) users

IBM Spectrum Scale supports OpenStack Swift and S3 protocols for object data access. Both of these object storage protocols use the Keystone service for identity management, and access by the object users to the object storage projects is controlled by these items:

- User roles
- Container ACLs

4.2.1 OpenStack Swift ACLs

Based on the roles that are defined for the user, object users can be administrative users or non-administrative users. Non-administrative users can perform operations only per container based on the container ACL headers:

- X-Container-Read
- X-Container-Write

Container ACLs are used to limit access to objects in Swift containers. Read access can be limited to allow only download, or allow download and listing. Write access allows the user to upload new objects to a container.

To learn about steps and see examples of creating container and importance of roles, see IBM Knowledge Center:

`http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_createcontainer.htm`

A Keystone administrator can create a container and grant `read` (ACL) permissions by using X-Container-Read headers. To learn about the steps and see examples of creating container with X-Container-Read headers, see IBM Knowledge Center:

`http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_createreadacl.htm`

Table 4-1 lists various ACL options that are available to manipulate object read ACLs.

Table 4-1 ACL options

Permission	Read ACL options
Read for all referrers	.r:*
Read and list for all referrers and listing	.r:*,.rlistings
Read and list for a user in a specific project	<project_name|project_id>:<user_name|user_id>
Read and list for a user in every project	*:<user_name|user_id>
Read and list for every user in a project	<project_name|project_id>:<*>
Read and list for every user in every project	<*>:<*>

A Keystone administrator can create a container and grant write (ACL) permissions by using X-Container-Write headers. To learn about steps and see examples of creating container with X-Container-Write headers, see IBM Knowledge Center:

`http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_writeacls.htm`

Note: Use a comma (,) in the header value to separate multiple ACLs.

4.2.2 OpenStack Swift3 ACLs

IBM Spectrum Scale uses Swift3 Middleware for OpenStack Swift, allowing access to IBM Spectrum Scale by using the Amazon Simple Storage Service (S3) API. To learn about steps to configure Swift3 middleware, see IBM Knowledge Center:

`http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_ChangeconfigurationenableS3.htm`

The credentials that are used by Amazon S3 and Elastic Compute Cloud (EC2) APIs are different from the credentials that are used by the OpenStack API. As a result, the user must generate these special credentials to use them when accessing the IBM Spectrum Scale OpenStack services. To learn about steps to configure EC2 credentials, see IBM Knowledge Center:

`http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_ConfigureOpenstackEC2credentials.htm`

IBM Spectrum Scale supports S3 ACLs on buckets and objects. These S3 ACLs are stored separately from the ACLs set through the Swift API and the ACLs that are stored in the file system (NFSv4 or POSIX).

To learn about managing OpenStack ACLs by using S3 API emulation, see IBM Knowledge Center:

`http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_ManagingOpenStackACLsviaAmazonS3API.htm`

To learn about using S3 API with s3curl, see the following website:

https://ibm.biz/BdspMV

4.2.3 Recommendations for ACL usage

When using ACLs for a larger set of files, here are some preferred practices that are generated by typical scenarios:

- If the access to the data includes access from Microsoft Windows client systems over the SMB protocol, then the ACL administration should be done from a Microsoft Windows client system. This technique ensures that the additional ACL flags that are used by Microsoft Windows are present, and that the order of the ACL entries is also the expected "canonical" order.
- Use the inheritance flags to have permissions that are inherited down from the root directory of an export.
- Avoid deny entries because they cause additional complexity that can be avoided.
- To be aware of product-specific limitations in the handling of ACLs, see IBM Knowledge Center:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_fileauthlimitations.htm
- To learn about known limitations for S3 API support, see IBM Knowledge Center:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_ManagingOpenStackACLsviaAmazonS3API.htm

When implementing unified file and object access where the same data can be accessed by using the Object interface and the File interface, the authorization of data is governed by the authorization semantics of the interface that the user is using to access the data. If the user is accessing the data by using SMB, then the SMB ACL is enforced on that data. However, when the same user accesses the same data by using the object interface, the object ACL is enforced.

4.2.4 Access control in IBM Spectrum Scale object

The following sections describe access control in IBM Spectrum Scale object.

Enabling non-admin users to perform Swift operations

A keystone administrator is defined with an `admin` role and can perform identity operations, such as user/project creation, endpoint listing, and so on. An IBM Spectrum Scale object user is not required to have administrative privileges. However, the user can be enabled to perform Swift operations by creating a new role, such as `role1`, and updating proxy-server.conf using the **`mmobj config change`** command as follows:

```
mmobj config change –ccrfile proxy-server.conf –section filter:keystoneauth
–property operator_roles –value "admin, SwiftOperator, role1"
```

Before running the **`mmobj config change`** command, the `proxy-server.conf` keystoneauth section will be as follows:

```
[filter:keystoneauth] reseller_admin_role = ResellerAdmin
use = egg:swift#keystoneauth
operator_roles = admin, SwiftOperator
is_admin = true
cache = swift.cache
```

After running the **mmobj config change** command, the proxy-server.conf keystoneauth section will be updated as follows:

```
[filter:keystoneauth] reseller_admin_role = ResellerAdmin
use = egg:swift#keystoneauth
operator_roles = admin, SwiftOperator, role1
is_admin = true
cache = swift.cache
```

The user can now perform IBM Spectrum Scale object operations using role1 but will not be able to perform keystone operations.

IBM Spectrum Scale Object is based on OpenStack Swift and Keystone. For more information on understanding Projects, Roles, and Users in the context of IBM Spectrum Scale object, see the following link:

https://docs.openstack.org/mitaka/install-guide-obs/keystone-users.html

Types of role-based authorization supported in IBM Spectrum Scale

There are two types of object roles in IBM Spectrum Scale, as defined in proxy-server.conf:

1. Reseller admin role: Any role defined as Reseller admin role will have control to all projects and containers/objects.
2. Operator role: Users assigned to an operator role will have access to all containers within the project that the user belongs to. To access containers from other projects, the user needs to be granted specific container ACLs.

Note: To enable a Keystone user to operate on Swift, the user must be assigned to either of these roles as it is not done by default.

Note: IBM Spectrum Scale uses KeystoneAuth that does not support account level ACLs.

Restricting container access for users within the same project

All users with an operator role within a project have access to all of the containers within that project.

To restrict access to a container, the user needs to be removed from that project where the container has been created. You can add the user to a different or a new project.

For more information about how to create a new project, see the following link:

https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_mngobjusers.htm

Granting read/write access to users from other projects

In various scenarios, when data from a container belonging to one project needs to be shared with users belonging to other projects, container ACLs need to be set.

A Swift user with an operator role can create container ACLs to grant read permissions using X-Container-Read headers in the curl tool or –read-acl flag in the Swift Command Line Client and X-Container-Write headers in the curl tool or –write-acl flag in the Swift Command Line Client to grant write permissions.

For example, there are u1, u2, u3 users in project1 and an u4 user in project2. The u4 user has created a test_bucket container in project2. The Swift user with an operator role in project2 has set the following ACLs on the container to allow read and write access to the u1 user, and read-only access to the u2 user. The u3 user will continue to have no access because the user belongs to a different project and no explicit ACLs have been set for it:

```
swift post test_bucket -r “project1:u1,project1:u2? -w “project1:u1”
```

To allow read access to all users in project1:

```
swift post test_bucket -r "project1:*"
```

To allow read access to all authenticated users from any project:

```
swift post test_bucket -r "*:*"
```

For more information on setting ACLs, see the following links:

- https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_configobjacls.htm
- https://docs.openstack.org/swift/latest/overview_acl.html

Allowing anonymous access to a container

In some scenarios, anonymous read access is required for certain objects. To allow read access for objects in a container to everyone without requiring authentication, set the following ACL:

```
swift post -r ".r:*" container_name
```

To allow everyone to do the container listing, set the following ACL:

```
swift post -r ".r:*,.rlistings" container_name
```

Now it should be possible to use a REST API client to read objects and perform container listing without authentication.

For example, a `curl` request to read an object will be as follows:

```
[root@srnode1 ~]# curl -i
http://srnode1:8080/v1/AUTH_8375ad77be424a44be183945f4c3d5f6/smitalc1/file1 -X GET
HTTP/1.1 200 OK
Content-Length: 0
X-Object-Meta-Objectization-Timestamp: 1516095881.92235
Accept-Ranges: bytes
Last-Modified: Tue, 16 Jan 2018 09:41:45 GMT
Etag: d41d8cd98f00b204e9800998ecf8427e
X-Timestamp: 1516095704.41961
Content-Type: application/octet-stream
X-Trans-Id: tx2b03950afbd34df0ab364-005a70171b
Date: Tue, 30 Jan 2018 06:56:27 GMT
```

Note: The Swift client cannot be used for performing object I/O with anonymous access, because the client uses the auth token.

Creating a superuser to allow operations on any account

In various scenarios where a user is required to operate on all IBM Spectrum Scale object user accounts, a reseller admin role can be defined in the proxy-server.conf file, using the **`mmobj config change`** command.

Complete the following steps:

1. Create the role:

```
[filter:keystoneauth]
reseller_admin_role = ResellerAdmin
use = egg:swift#keystoneauth
operator_roles = admin, SwiftOperator, role1, role2, role3
is_admin = true
```

2. Create the role of `ResellerAdmin` on the IBM Spectrum Scale cluster, if it does not exist:

```
[root@srnode1 ~]# openstack role create ResellerAdmin
+——–+——————————-+
| Field | Value |
+——–+——————————-+
| domain_id | None |
| id | 1d5eceeb7d614e678e2b930e83f10ff2 |
| name | ResellerAdmin |
+——–+——————————-+
```

3. Assign the role to a user who should be the superuser:

```
[root@srnode1 ~]# openstack role add –project proj1 –user u1 ResellerAdmin
```

4. The user u1 can now do any operations on any projects, for example:

```
[root@srnode1 ~]# swift list
—os-storage-url=’http://localhost:8080/v1/AUTH_e6030971a9424d739d281b4cf6b46939
?
a1_test_bucket
cont1
test_bucket
[root@srnode1 ~]# swift list cont1
—os-storage-url=’http://localhost:8080/v1/AUTH_e6030971a9424d739d281b4cf6b46939
?
s3curl.pl
[root@srnode1 ~]# swift upload cont1 openrc.u4
—os-storage-url=’http://localhost:8080/v1/AUTH_e6030971a9424d739d281b4cf6b46939
?
openrc.u4
```

For more information about reseller admin, see the following link:

https://www.ibm.com/developerworks/community/forums/html/topic?id=813eb4fd-c0c6-43a5-9317-a35e4081ef72

4.3 References

The following websites are useful for further research:

- Creating containers:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_createcontainer.htm

- Creating read ACLs to authorize object users:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_createreadacl.htm

- Creating write ACLs to authorize object users:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_writeacls.htm

- Changing the object base configuration to enable S3 API:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_ChangeconfigurationenableS3.htm

- Configuring OpenStack EC2 credentials:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_ConfigureOpenstackEC2credentials.htm

- Managing OpenStack ACLs by using the S3 API:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_ManagingOpenStackACLsviaAmazonS3API.htm

- Getting started with the IBM Spectrum Scale S3 API support by using s3curl:

  https://ibm.biz/BdspMV

- Authorization limitations:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_fileauthlimitations.htm

- Managing OpenStack ACLs by using S3 API:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_ManagingOpenStackACLsviaAmazonS3API.htm

5

Secure administration

Given the distributed (multi-node and multi-cluster) nature of IBM Spectrum Scale, cluster administration involves tasks where commands and data flow across node boundaries. Secure cluster administration includes consideration for one operating system instance and on how that node communicates with others.

This chapter includes the following sections:

- Remote Shell and Remote Copy
- Running IBM Spectrum Scale without remote root login
- Secure administration by using the GUI
- Secure administration by using the REST API
- References

5.1 Remote Shell and Remote Copy

Root authority is required to perform all IBM Spectrum Scale administration tasks except for the ones with a function that is limited to listing certain IBM Spectrum Scale operating characteristics or modifying individual user file attributes.

The IBM Spectrum Scale commands maintain the environment across all nodes in the cluster. To achieve this goal, the IBM Spectrum Scale commands use the *remote shell* and *remote file copy* commands that are specified during cluster creation (the **mmcrcluster** command).

The default remote commands are **ssh** and **scp**, but any other remote commands can be designated if they have compatible syntax.

In principle, IBM Spectrum Scale administration commands can be issued from any node in the cluster. The nodes that are planned to be used for administering IBM Spectrum Scale must be able to run remote shell commands on themselves and on any other node in the cluster. They must do so without the use of a password and without producing any extraneous messages. Similarly, the nodes on which the IBM Spectrum Scale commands are issued must be able to copy files to and from any other node in the cluster. The nodes must also do so without the use of a password and without producing any extraneous messages.

The way the password-less access is achieved depends on the remote execution program and authentication mechanism that is used. If the remote program is **ssh**, you can use private identity files that do not have a password. Or, if the identity file is password-protected, you can use the **ssh-agent** utility to establish an authorized session before you issue **mm** commands.

The **adminMode** configuration attribute specifies whether all nodes in the cluster can be used for issuing IBM Spectrum Scale administration commands or just a subset of the nodes:

allToAll Indicates that all nodes in the cluster can be used for running IBM Spectrum Scale administration commands, and that all nodes can run remote commands on any other node in the cluster without needing a password.

central Indicates that only a subset of the nodes can be used for running IBM Spectrum Scale commands, and that only those nodes can run remote commands on the rest of the nodes in the cluster without needing a password. This mode is the default (starting with IBM Spectrum Scale V3.5).

The major advantage of the **central** mode of administration is that the number of nodes that must have root level access to the rest of the nodes is limited, and can be as low as one. If password-less **ssh** is set up, then only the nodes from which the commands are issued must have password-less **ssh** access to other nodes in the cluster.

The disadvantage is that IBM Spectrum Scale might not be able to automatically recover from the loss of certain configuration files. For example, if the SSL key files (which are used to allow authentication of file system daemon RPCs) are not present on some of the nodes, the operator might have to intervene to recover the missing data. Similarly, it might be necessary to shut down IBM Spectrum Scale when adding quorum nodes.

IBM Spectrum Scale does not know which nodes are being used for administration purposes, and it does not limit from which nodes the commands can be issued. It is the administrator's responsibility to issue the administration commands only from nodes that are properly configured and can access the rest of the nodes in the cluster. IBM Spectrum Scale does not track which nodes are used for administration. Issuing commands from nodes that are not enabled for remote shell to other nodes in the cluster might result in those commands failing.

The nodes where the GUI is installed are explicitly chosen as a point of administration. Those nodes must be able to start remote shell functions.

5.2 Running IBM Spectrum Scale without remote root login

In several environments, corporate IT policies require that the **`ssh PermitRootLogin`** parameter is disabled to prevent remote login as root. That requirement does not prevent the cluster from being successfully administered.

By using **`sudo`** and the IBM Spectrum Scale **`sudo`** wrappers, IBM Spectrum Scale administration can be performed securely by using a non-root ID. The root user on a node that is used for administration still must be able to log in to all nodes in the cluster as the non-root ID, without being prompted for a password.

The IBM Spectrum Scale **`sudo`** wrappers that enable IBM Spectrum Scale administrative operations to be securely performed by using a non-root user rely on **`ssh`** wrappers to start remote commands with a non-root user ID, and then use **`sudo`** on the remote node to run the necessary commands.

The high-level setup involves the following tasks:

1. Configure **`sudo`**.
2. Configure the IBM Spectrum Scale cluster to use **`sudo`** wrapper scripts.

Configuring **`sudo`** requires installing **`sudo`** across the cluster and also these steps:

1. Create a user and group with **`sudo`** privileges to run the IBM Spectrum Scale administration commands, for example, the non-root user name `gpfsadmin` and the group `gpfs`.
2. Allow password-less access to the root user from any IBM Spectrum Scale administration node to issue (remote) commands on all nodes with the user ID `gpfsadmin`.
3. Edit the `/etc/sudoers` file in each node of the cluster to allow some commands to be run by members of the `gpfs` group. A few of them, especially `/usr/lpp/mmfs/bin/mmremote`, should be run without requiring a password.

After the initial **`sudo`** setup, the cluster should be configured to operate with the **`sudo`** wrapper scripts **`sshwrap`** and **`scpwrap`** (instead of **`ssh`** and **`scp`**), which are included with the product. The wrappers can be configured at cluster creation time by specifying the **`--use-sudo-wrapper`** option of the **`mmcrcluster`** command, as shown in Example 5-1.

Example 5-1 The mmcrcluster --use-sudo-wrapper command

```
$ whoami
gpfsadmin
$ sudo /usr/lpp/mmfs/bin/mmcrcluster -N mmcrcluster-node.lst --use-sudo-wrapper -C
gpfsSudoTest
```

After the cluster is created, other commands can be issued by the `gpfsadmin` user by passing the command names and parameters to the **`sudo`** program, for example:

```
$ sudo /usr/lpp/mmfs/bin/mmcrnsd -F nsdStanzaFile
```

Existing clusters can be changed to operate with the wrappers by using the **`--use-sudo-wrapper`** option with **`mmchcluster`**.

To use the IBM Spectrum Scale management GUI on a cluster where **sudo** wrappers are used, issue the **mmchconfig sudoUser=gpfsadmin** command to configure the user name, and then restart the GUI using the **systemctl restart gpfsgui** command.

The IBM Spectrum Scale commands that are run under **sudo** are logged in system files (/var/log/secure, for example), which is useful for auditing purposes. The command name, the issuer's user name, and the current directory are logged.

For more information, see IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_sudowrapper.htm

For more information about how to configure the GUI to use **sudo** wrappers, see IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_configguiforsudo.htm

5.3 Secure administration by using the GUI

IBM Spectrum Scale comes with a secure graphical user interface (GUI) for administration. Here are the key security aspects of the IBM Spectrum Scale GUI:

- Role-based access control for administration by using the GUI

  The IBM Spectrum Scale GUI supports different administrative roles. These predefined roles are associated with user groups that define the working scope within the GUI. The GUI comes with a default set of nine roles and nine user groups that are associated with the predefined roles. A user can be part of multiple user groups so that a single user can play multiple roles in the system. This feature enables division of responsibilities among multiple administrators based on roles.

  The GUI allows users to be local users. The default GUI user named *admin* can enforce strong passwords for the local users using the Password Policy tab in the GUI Users page.

  The GUI also allows users from central directory services, such as Microsoft Active Directory (AD) or LDAP.

  For more information, see IBM Knowledge Center:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_manageguiadmins.htm

- Support for sudo wrappers

  The IBM Spectrum Scale GUI supports sudo-based administration.

- Secure administration by using the GUI

  IBM Spectrum Scale supports secure access to the GUI by using `https` with the support for self-signed or trusted certificate authority (CA). For more information, see IBM Knowledge Center:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_managecertforgui.htm

- Root privilege considerations for IBM Spectrum Scale management GUI

  From the 4.2.3 release, the IBM Spectrum Scale GUI WebSphere® Java process no longer runs as `root` but as a user named `scalemgmt`. This change provides improved security because web applications running as `root` are vulnerable to security threats. The `scalemgmt` user is set up as a system account without any login privileges.

On installation, GUI adds the file /etc/sudoers.d/scalemgmt_sudoers that allows the scalemgmt user to run commands matching the wildcard "/usr/lpp/mmfs/bin/mm".

For more information, see IBM Knowledge Center:

https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1ins_guiconfignonrootprivileges.htm

- Remote Cluster monitoring using the IBM Spectrum Scale management GUI

  From the 5.0.0 release, GUI supports remote cluster monitoring to display the performance of the remote cluster with the help of performance monitoring tools that are configured in both the remote and local clusters.

  The remote cluster monitoring uses HTTPS for communication between the two GUIs and a token-based authentication similar to the local user authentication used by a single GUI. Only read access is allowed, and therefore an administrator cannot modify any values on the remote GUI.

  For information about monitoring the performance of a remote cluster, see IBM Knowledge Center:

  https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1hlp_accessremoteconnections.htm

  The GUI also supports remote cluster monitoring to display AFM-related data. This option is available under **Access** > **Remote Connections**. For example, a user could view performance data from the AFM cache cluster in the GUI running on the AFM home cluster.

  For information about monitoring the AFM and AFM DR configuration, health, and performance across clusters, see the following link:

  https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_guimonafm.htm

  To access the REST API service of the GUI in the remote cluster, an authentication token is used. Both GUIs need to communicate using port 443. Access can be revoked at any time and only read access is allowed.

5.4 Secure administration by using the REST API

The following section explains the two versions of REST API and REST API handling of Cross-Origin Resource Sharing (CORS).

5.4.1 REST API version 1

IBM Spectrum Scale V4.2.2.x supports the REST-based API for management of some of the key administration functions, such as filesets, snapshots, and quotas. These REST APIs provide secure administration:

- Mandating authenticated requests
- Supporting role-based access control (RBAC)
- Ensuring secure administration over the wire by leveraging SSL/TLS channels
- Supporting IBM Spectrum Scale **sudo** wrappers for deployments for secure administration with non-root remote credentials.

The audit log maintains a record of various actions that are performed on the system. This helps the system administrator to audit the commands and tasks that the users and administrators are performing. These logs can also be used to troubleshoot issues that are reported in the system.

The REST API supports the use of RBAC that is available when IBM Spectrum Scale is managed by the GUI. In the 4.2.2 release, when using the IBM Spectrum Scale GUI as an authentication back end for management, the roles that are defined in the IBM Spectrum Scale management GUI are used (For more information, see 5.3, “Secure administration by using the GUI” on page 36).

As described in 5.2, “Running IBM Spectrum Scale without remote root login” on page 35, there are deployments where you may choose to use **sudo**-based administration for increased security. The IBM Spectrum Scale REST API also supports **sudo**-based administration.

Note: From IBM Spectrum Scale V4.2.3, the use of REST API version 1 has been deprecated.

5.4.2 REST API version 2

Starting with IBM Spectrum Scale V4.2.3, the REST-ful management API is hosted by the IBM Spectrum Scale GUI software stack, and is not running as a separate daemon. The GUI web server is the backend server for every REST API call and therefore, all authorization and authentication mechanisms supported by the GUI are valid for REST API because it is the same software stack. Encryption and other security-related considerations are similar to the GUI. For more information, see 5.3, “Secure administration by using the GUI” on page 36.

The IBM Spectrum Scale REST API supports **sudo**-based administration. Therefore, for increased security, you can choose to use **sudo**-based administration in some deployments, as described in 5.2, “Running IBM Spectrum Scale without remote root login” on page 35.

5.4.3 REST API and Cross-Origin Resource Sharing (CORS)

Calls to the REST API are only allowed through HTTPS. The REST API sets the following headers on every response to handle Cross-Origin Resource Sharing (CORS):

- Access-Control-Allow-Origin: set to the `Origin` header of the request
- Access-Control-Allow-Credentials: `true`
- Access-Control-Allow-Methods: `GET`, `POST`, `PUT`, `DELETE`
- Access-Control-Allow-Headers: `Authorization`, `Content-Type`, `Accept`

Any call to the REST API using the HTTP method `OPTIONS` returns the previous headers and no payload to allow browsers to set the correct CORS headers on every subsequent call.

For more information about IBM Spectrum Scale management API, see IBM Knowledge Center:

https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_restapi_main.htm

5.5 References

The following websites are useful for more detailed research:

- *GPFS and Remote Shell*:

  https://ibm.biz/BdspS2

- Running IBM Spectrum Scale without remote root login:

  http://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_sudowrapper.htm

- Managing GUI administrators:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_manageguiadmins.htm

- The adminMode configuration attribute:

  https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_adminmodeparm.htm

- IBM Spectrum Scale management API:

  https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_restapi_main.htm

6

Immutability

Tamper-proof data is ensured by the immutability feature, which falls in the storage system touch point *Storing of Data*.

IBM Spectrum Scale immutability is based on *immutable filesets*. Immutable filesets enable managing immutable files similar to the SnapLock method from NetApp Inc. Immutable filesets can be exported using the Network File System (NFS) protocol and Server Message Block (SMB) protocol. This feature makes it easy for applications supporting the SnapLock semantic by using NFS and SMB to adopt IBM Spectrum Scale as an immutable file storage target.

In an immutable fileset, files can be immutable or append-only for a configurable retention time by using standard file system commands. During the retention time, immutable files cannot be deleted or modified. When the retention time expires, immutable files can be deleted but cannot be modified.

With the immutability function, IBM Spectrum Scale can be used for archiving use cases where regulatory requirements demand that the implementation prevent modification and deletion of files. To underline the compliance aspect, IBM assessed this function in accordance with US, German, and Swiss laws and regulations, and engaged an independent and worldwide recognized auditor.

For information and use cases of IBM Spectrum Scale immutability, see the *IBM Spectrum Scale Immutability Introduction, Configuration Guidance, and Use Cases*, REDP-5507:

http://www.redbooks.ibm.com/abstracts/redp5507.html

This chapter includes the following sections:

- IBM Spectrum Scale as an archive storage
- Immutable filesets
- References

6.1 IBM Spectrum Scale as an archive storage

Archiving is characterized by medium-to-large volumes of data that must be kept for long periods. Certain types of data, such as trade and tax records, must be kept in an immutable manner according to laws and regulations. Due to the long archiving lifetimes, it is important to manage cost for these operations, because the files should always be accessible.

IBM Spectrum Scale provides a comprehensive set of functions that are made for archiving:

- High availability of file systems services and files across sites through synchronous replication and reliable quorum techniques. This function allows continuous operations even during a site outage.
- Disaster protection and recovery ensure that if a disaster occurs, files can be recovered from a remote site, either by using backup techniques or asynchronous replication. This function facilitates instant failover and recovery.
- Tiered storage enables transparently placing files on the most appropriate storage medium during the lifecycle. IBM Spectrum Scale supports many types of storage media, including flash, disk, and tape. This function enables optimization of storage cost, for example, by moving files that are no longer accessed but must be retained for many years to tape. Tape provides 5 - 10 times lower total cost of ownership (TCO) than disk. For more information, see *Clipper Group: Disk and tape total cost of ownership study*:

  `http://www.clipper.com/research/TCG2015006.pdf`
- Immutability and encryption for all or subsets of files prevent data tampering and provides confidentiality. They also help to comply with legal requirements and business standards.
- With command audit logging, administrative activities in IBM Spectrum Scale can be recorded and made available for subsequent audits.
- File audit logging monitors the file operations and provides audit trails that are required by certain laws and regulations.
- IBM Spectrum Scale allows secure administration with named administrative users through **`sudo`** wrappers. This eliminates the need of the `root` user for administrative activities.

6.2 Immutable filesets

An IBM Spectrum Scale fileset is a partition in a file system that is seen as a directory from a user perspective. Certain functions can be configured on a fileset level, such as immutability.

IBM Spectrum Scale supports one of the following immutability (`IAM`) modes for an immutable fileset:

- None: No immutability mode is set (default). The fileset is a regular fileset.
- Advisory: Enables setting retention times and immutability, but files can be deleted with the proper file permission.
- Noncompliant: Advisory mode plus files cannot be deleted if retention time has not yet expired. However, retention times can be reset, and files can be deleted but not changed.
- Compliant: Noncompliant mode plus retention time cannot be reset. When retention time expires, files can be deleted but not changed.

The immutability mode on a fileset can be upgraded from `advisory` to `noncompliant` to `compliant`, but not downgraded.

The following example demonstrates how to configure an immutable fileset in an IBM Spectrum Scale file system:

1. Create a fileset in the file system by using the following command:

   ```
   # mmcrfileset filesystem-name fileset-name --inode-space new
   ```

2. The fileset must be linked to a directory in the file system by using the following command:

   ```
   # mmlinkfileset filesystem-name fileset-name -J junction-path
   ```

3. Now, the immutability mode can be set. In this example, set the mode to `compliant` by using the following command:

   ```
   # mmchfileset filesystem-name fileset --iam-mode compliant
   ```

4. To list the `IAM` mode (immutability) of a fileset, use the following command:

   ```
   # mmlsfileset filesystem-name fileset --iam-mode
   ```

Files that are stored in an immutable fileset can be set to `immutable`. Setting files to `immutable` involves two steps:

1. Setting the retention time of the file.
2. Setting the file to `immutable`.

These two steps must be completed for every file that is stored in an immutable fileset to make the file immutable. If a file is not processed this way, it remains a normal file that can be modified and deleted.

File immutability can be managed with standard commands that are available in UNIX systems (for example, **`chmod`** and **`chattr`**), with specific IBM Spectrum Scale commands, or by using SMB that uses Microsoft Windows PowerShell, as described in the following references:

- *Managing immutability with IBM Spectrum Scale commands*:

  https://www.ibm.com/developerworks/community/blogs/storageneers/entry/Insight_to_the_IBM_Spectrum_Scale_GPFS_Immutability_function?lang=en

- *Managing immutability by using NFS and SMB exports*:

  https://ibm.biz/BdspSp

6.3 References

The following websites are useful for further research:

- *IBM Spectrum Scale Immutability Introduction, Configuration Guidance, and Use Cases*, REDP-5507:

  http://www.redbooks.ibm.com/abstracts/redp5507.html

- Blog: *Insight to the IBM Spectrum Scale (GPFS) immutability function*:

  https://www.ibm.com/developerworks/community/blogs/storageneers/entry/Insight_to_the_IBM_Spectrum_Scale_GPFS_Immutability_function?lang=en

- Blog about managing immutability by using NFS exports and SMB shares:

  https://ibm.biz/BdspSp

- End-to-End checksums with IBM Spectrum Scale Native RAID:

  http://www.ibm.com/support/knowledgecenter/en/SSYSP8_4.5.0/com.ibm.spectrum.scale.raid.v4r21.adm.doc/bl1adv_introe2echecksum.htm

- Configuring sudo-wrappers in an IBM Spectrum Scale cluster:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_sudowrapper.htm

- Clipper Group: Disk and tape total cost of ownership study:

  http://www.clipper.com/research/TCG2015006.pdf

- SEC Rule 17a-4(f) Assessment Report:

  http://www.kpmg.de/bescheinigungen/RequestReport.aspx?41742

- KPMG Software certificate:

  https://www.kpmg.de/bescheinigungen/RequestReport.aspx?41743

7

Audit logging

Auditing file system activities and administrative operations is an important security aspect in a number of deployments. With the advent of industry and government regulations, there are now requirements to log certain file system activities, such as file creation, deletion, modification, and renaming.

This chapter describes how the file audit logging feature, introduced in IBM Spectrum Scale 5.0.0 release, captures file operations occurring on an audited file system and logs them into a retention enabled fileset.

Varonis DatAdvantage can be used to audit file system activities for releases prior to IBM Spectrum Scale 5.0.0 release, for more information, see the “References” on page 54.

In addition, the chapter presents the facility in IBM Spectrum Scale to audit cluster configuration changes.

This chapter includes the following sections:

- File audit logging
- Audit logging for cluster configuration changes
- References

7.1 File audit logging

File audit logging captures file operations on a file system. When files are accessed on a node, a corresponding event is generated for the access. Each node publishes its events to a distributed queue. Listeners subscribe to the queue, dequeue the events, and then log the events to an immutable repository.

The events, also called *Light Weight Events*, have been integrated into the IBM Spectrum Scale policy engine since the 5.0.0 release.

File Audit Logging is available under the IBM Spectrum Scale Advanced Edition and Data Management Edition.

From a file system administrator's perspective, IBM Spectrum Scale file audit logging consists of the following components configured when file audit logging is installed.

- Producers

  This component is activated on all the nodes and publishes events to the distributed queue.

- Distributed Message Queue

  This component is installed on a set of nodes (selected by the administrator). The message queue collects and stores events that are published by the producers. The messages are then drained by consumers. The message queue is made up of Kafka brokers also called *Message Queue Servers*.

- Consumers

  This component is installed on all the nodes where Message Queue Server exists. The consumers subscribe to the queue listening for events being published to the queue. When new events come in, the consumers drain the events off of the queue and persist them to a sink.

  In IBM Spectrum Scale 5.0.0, file audit logs are written to a tamper-proof fileset. A group of consumers, called a Consumer group, cluster together to provide for scalability and availability. All consumers subscribing to a file system's events cluster together to form a consumer group. For more information, see IBM Spectrum Scale Knowledge Center:

  https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1ins_adlgconsumers.htm

- Topics

  The queue is divided into sections called *topics*. Topics allow a queue to be shared and used for multiple purposes. From IBM Spectrum Scale 5.0.0, to share the distributed queue across multiple file systems that need to be audited, a unique topic is created for each such file system. Producers that publish events for a file system do so to a unique topic on the queue created specifically for that file system. Like producers, consumers subscribe to messages for a specific file system. Therefore, they subscribe to the topic designated for that file system.

- File Audit Logging Repository / Audit Logging Fileset

  Each audited file system is designated a fileset where its audit logs are persisted. The default fileset is `.audit_log`, though this is configurable. This fileset keeps the logs currently being written to in an append-only mode. When a predetermined number of logs are written to a file, the file is closed, compressed, made immutable, and also assigned a retention period. Therefore, new audit files are created at a rate that corresponds to the workload driving the file system.

File audit logging records for a given file system are logged in files that are arranged in the following manner:

```
<FS_Mount_Point>/<Fileset_Name>/Topic/Year/Month/Day
```

Files in the file audit logging fileset are named in the following format:

```
auditLogFile_<Consumer_Node_Name>_YYYY-MM-DD_HH:MM:SS
```

- File audit logging records

  The following file access operations are logged: open, close, destroy (delete), rename, unlink, create, remove directory, extended attribute change, and ACL change. Events are JSON formatted. You can use tools or scripts if these logs need to be forwarded to an external audit server based on business needs. Example 7-1 is a sample of an audit record.

Example 7-1 Sample of an audit record

```
{“LWE_JSON”: “0.0.1”, “path”: “/ibm/fs1/one”, “oldPath”: null, “clusterName”:
“gpfs.scalecluster”, “nodeName”: “ces1”, “nfsClientIp”: “”, “fsName”: “fs1”,
“event”: “CREATE”, “inode”: “95488”, “linkCount”: “1”, “openFlags”: “0”,
“poolName”: “system”, “fileSize”: “0”, “ownerUserId”: “0”, “ownerGroupId”: “0”,
“atime”: “2018-06-12_16:21:36+0530”, “ctime”: “2018-06-12_16:21:36+0530”,
“eventTime”: “2018-06-12_16:21:36+0530”, “clientUserId”: “0”, “clientGroupId”:
“0”, “processId”: “32210”, “permissions”: “200100644”, “acls”: null, “xattrs”:
null, “subEvent”: “NONE” }
```

> **Note:** File audit logging is enabled at the file system level. Therefore, you are required to plan the file system, which needs to be enabled for logging, based on the data it hosts and the business or compliance requirements.

Figure 7-1 shows an IBM Spectrum Scale cluster with an illustration of where the components are installed and how they interact. For simplicity, neither consumer group or message queue topics is shown in the figure.

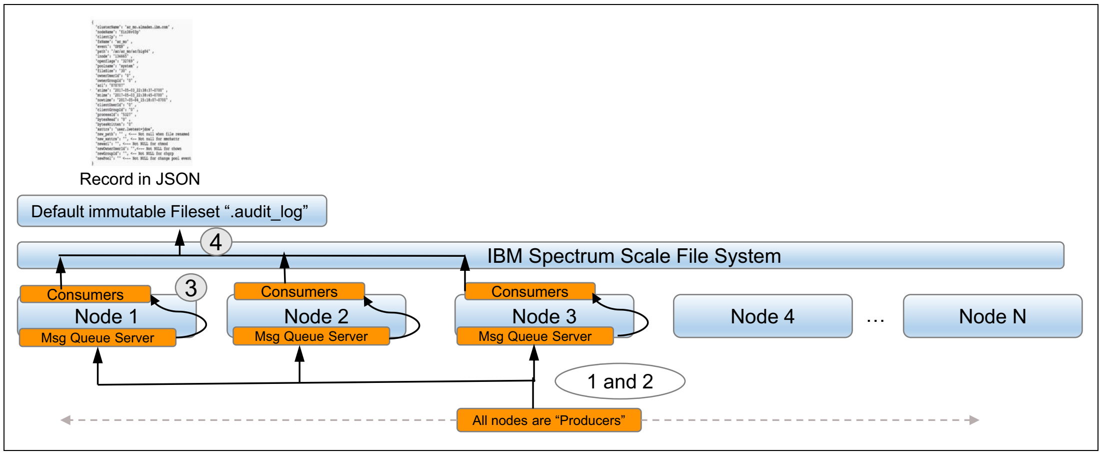


Figure 7-1 IBM Spectrum Scale File Audit Logging Components and High Level Flow

- **Step 1**: On every file operation on any given node (which includes SMB, NFS, POSIX interface), an event encapsulating the details of the operation is generated by the producer.
- **Step 2**: The event generated by the producer is delivered to the Message Queue Server (to a specific topic).

- **Step 3**: Consumers (subscribing to a specific topic) drain these events from the message queue.
- **Step 4**: Consumers write the events into the immutable fileset on the file system (".audit_log" being the default one)

> **Note:** IBM Spectrum Scale File Audit logging uses Apache Kafka as its distributed streaming platform, so concepts and terminology are similar to it.

7.1.1 Installation and enablement

File audit logging feature can be installed using the IBM Spectrum Scale installation toolkit to enable and configure `FAL` in the cluster definition file. See Figure 7-2.

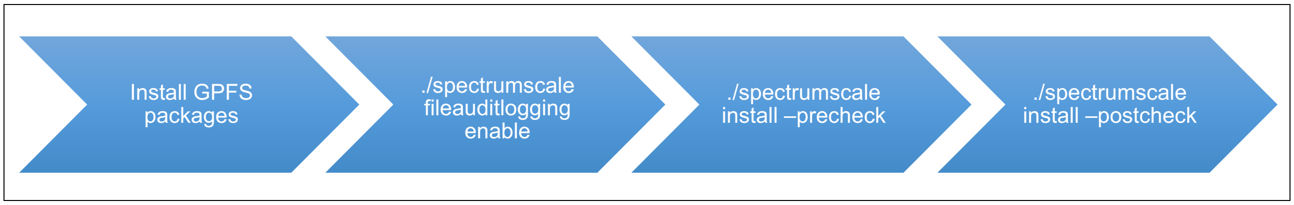

Install GPFS packages
./spectrumscale fileauditlogging enable
./spectrumscale install –precheck
./spectrumscale install –postcheck

Figure 7-2 Process to enable and configure FAL in the cluster definition file

After enabling the feature at the cluster level, it needs to be enabled for each file system. See Figure 7-3.

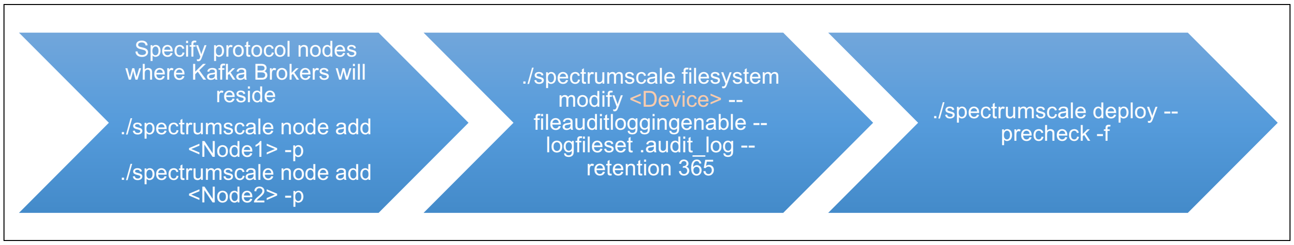

Specify protocol nodes where Kafka Brokers will reside
./spectrumscale node add <Node1> -p
./spectrumscale node add <Node2> -p
./spectrumscale filesystem modify <Device> --fileauditloggingenable --logfileset .audit_log --retention 365
./spectrumscale deploy --precheck -f

Figure 7-3 Process for enabling FAL on each file system

The file audit logging configuration-related changes become effective after the deployment procedure is completed.

The following sample example of installing and configuring file audit logging uses the installer. The example does not include the full output for the command execution, but explains the general setup and config using the toolkit.

```
# ./spectrumscale fileauditlogging enable
[ INFO  ] Enabling file audit logging in the cluster configuration file.
[ INFO  ] Tip :If all node designations and any required file audit logging
configurations are complete, proceed to assign filesystem to enable file audit
logging configuration: ./spectrumscale filesystem modify --fileauditloggingenable
<filesystem name>
```

> **Note:** It is a preferred practice to run **`./spectrumscale install --precheck`** and **`./spectrumscale install --postcheck`**.

```
# ./spectrumscale node add prt001st003 -p
[ INFO  ] Adding node prt001st003 as a GPFS node.
[ INFO  ] Setting prt001st003 as a protocol node.
[ INFO  ] Configuration updated.
[ INFO  ] Tip : If all node designations are complete, configure the protocol
environment as needed: ./spectrumscale config protocols -f cesSharedRoot -m
/ibm/cesSharedRoot

# ./spectrumscale node add prt002st003 -p
[ INFO  ] Adding node prt002st003 as a GPFS node.
[ INFO  ] Setting prt002st003 as a protocol node.
[ INFO  ] Configuration updated.
[ INFO  ] Tip : If all node designations are complete, configure the protocol
environment as needed: ./spectrumscale config protocols -f cesSharedRoot -m
/ibm/cesSharedRoot

# ./spectrumscale node add prt003st003 -p
[ INFO  ] Adding node prt003st003 as a GPFS node.
[ INFO  ] Setting prt003st003 as a protocol node.
[ INFO  ] Configuration updated.
[ INFO  ] Tip : If all node designations are complete, configure the protocol
environment as needed: ./spectrumscale config protocols -f cesSharedRoot -m
/ibm/cesSharedRoot
```

Note: If the deployment has less than three protocol nodes in their cluster, they must run **mmmsgqueue enable -N <atleast 3 nodes>**.

```
# ./spectrumscale filesystem modify --fileauditloggingenable gpfs0
[ INFO  ] The filesystem gpfs0 will be configured file audit logging with existing
.audit_log log fileset.
[ INFO  ] The filesystem gpfs0 will be configured file audit logging with existing
365 retention days.
[ INFO  ] The filesystem gpfs0 will be configured with file audit logging.
[ INFO  ] Tip : Now that you have modified this filesystem to use file audit
logging, you need to enable it using the './spectrumscale fileauditlogging enable'
command. please ignore if you have already enabled file audit logging.

# ./spectrumscale fileauditlogging list
[ INFO  ] File audit logging is Enabled.
[ INFO  ] User has defined Node ['prt001st003'] as broker node in the current
configuration.
[ INFO  ] Name                       Log-fileset name          Retention days
[ INFO  ] gpfs0                      .audit_log                365
```

You can change the audit fileset name and retention time here, but the defaults are set to 365 day retention, and the fileset named `.audit_log`.

```
# ./spectrumscale filesystem modify
usage: spectrumscale filesystem modify [-h]
                                       [-B {64K,128K,256K,512K,1M,2M,4M,8M,16M}]
                                       [-m MOUNTPOINT] [-r {1,2,3}]
                                       [-mr {1,2,3}] [-MR {1,2,3}]
                                       [-R {1,2,3}]
                                       [--metadata_block_size
{64K,128K,256K,512K,1M,2M,4M,8M,16M}]
                                       [--fileauditloggingenable]
                                       [--fileauditloggingdisable]
                                       [--logfileset LOGFILESET]
                                       [--log_fileset_device LOG_FILESET_DEVICE]
                                       [--retention RETENTION]
                                       filesystem
```

Before the deployment, you can check the node listings to see whether file audit logging has been enabled or not, using the **spectrumscale node list** command. On running the deployment, the prerequisite rpms (like KAFKA rpm) will be installed, the message queue will be set up (defaulting to use the protocol nodes as brokers), and file audit logging will be configured.

```
# ./spectrumscale deploy --precheck -f
[ INFO  ] Logging to file:
/usr/lpp/mmfs/5.0.2.0/installer/logs/DEPLOY-PRECHECK-10-08-2018_07:52:09.log
[ INFO  ] Validating configuration
…

# ./spectrumscale deploy
[ INFO  ] Logging to file:
/usr/lpp/mmfs/5.0.2.0/installer/logs/DEPLOY-09-08-2018_12:09:23.log
[ INFO  ] Validating configuration
      …
```

Manual installation

You can also manually install and configure IBM Spectrum Scale File Audit logging. For more information on the procedure, see the *Manually installing file audit logging* section in IBM Spectrum Scale Knowledge Center:

https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1ins_adlgmanualinstall.htm

The following example shows the steps required to install, enable and configure file audit logging for a given file system manually. The example has an IBM Spectrum Scale cluster with three protocol nodes already configured. You can view the cluster configuration using the **mmlscluster** command.

```
# mmlscluster
GPFS cluster information
========================
  GPFS cluster name:         scale.ibm.com
  GPFS cluster id:           11316316357699068975
  GPFS UID domain:           scale.ibm.com
  Remote shell command:      /usr/bin/ssh
  Remote file copy command:  /usr/bin/scp
  Repository type:           CCR
GPFS cluster configuration servers:
-----------------------------------
  Primary server:    nsd001st003 (not in use)
  Secondary server:  (none)
 Node  Daemon node name  IP address  Admin node name  Designation
------------------------------------------------------------------
   1   nsd001st003       11.11.0.2   nsd001st003      quorum-perfmon
   2   nsd002st003       11.11.0.1   nsd002st003      quorum-perfmon
   3   prt003st003       11.11.0.3   prt003st003
quorum-manager-gateway-perfmon
   4   prt002st003       11.11.0.4   prt002st003      manager-gateway-perfmon
   5   prt001st003       11.11.0.5   prt001st003      manager-gateway-perfmon
```

```
# mmlscluster --ces
GPFS cluster information
========================
  GPFS cluster name:         scale.ibm.com
  GPFS cluster id:           11316316357699068975
Cluster Export Services global parameters
-----------------------------------------
  Shared root directory:                /ibm/ces_fs0/ces
  Enabled Services:                     SMB NFS
  Log level:                            0
  Address distribution policy:          node-affinity
 Node  Daemon node name            IP address       CES IP address list
-----------------------------------------------------------------------
   3   prt003st003                 11.11.0.3        10.18.116.102  1.2.3.4
   4   prt002st003                 11.11.0.4        1.2.3.5
   5   prt001st003                 11.11.0.5        10.18.116.100 10.18.116.101
1.2.3.6
```

Install

To manually install and configure file audit logging, install the `gpfs.kafka-5.0*.rpm` and `gpfs.librdkafka-5.0*.rpm` packages on each node of the cluster.

Enable message queue

After the RPMs are installed, use the **mmmsgqueue** command to enable the message queue. If the cluster does not have at least three protocol nodes or you want to have a different set of nodes to be the message server, you must pass three nodes into the command with the **-N** flag. This step will make the message queue use eligible Linux quorum nodes as ZooKeepers, use the nodes specified by the **-N** option as message queue servers/brokers, and default to all the protocols nodes if you do not pass a **-N** flag.

The following command shows no node class associated with file audit logging defined by an administrator:

```
# mmlsnodeclass
Node Class Name       Members
--------------------- -----------------------------------------------------------
CALLHOME_SERVERS      prt003st003
CALLHOME_CLIENTS      nsd001st003,prt003st003,nsd002st003,prt002st003 prt001st003
GUI_MGMT_SERVERS      prt002st003,prt001st003
GUI_SERVERS           nsd001st003,prt003st003,nsd002st003,prt002st003 prt001st003
```

The following example shows enabling of message queue (no **-N** flag), which leads the **kafkaBrokerServers** node class to consist of all of the protocol nodes. Also note that the zookeepers are set up on the quorum nodes by default.

```
# mmmsgqueue enable
[I] The kafkaZookeeperServers node class was successfully created with 3 member
nodes.
[I] The kafkaBrokerServers node class was successfully created with 3 member
nodes.
[I] Successfully created Kafka broker configuration file and added to CCR.
[I] Successfully created Kafka Zookeeper configuration file and added to CCR.
[I] Enabling MsgQueue daemons.
[I] Creating callbacks to control starting and stopping the MsgQueue daemons.
[I] Pushing producer authentication information to eligible cluster nodes.
   Depending on cluster size, this may take some time.
```

```
[I] Enabling access to Spectrum Scale topics within the MsgQueue.
[I] MsgQueue successfully enabled.

# mmlsnodeclass
Node Class Name       Members
--------------------- -------------------------------------------------------
CALLHOME_SERVERS      prt003st003
CALLHOME_CLIENTS      nsd001st003,prt003st003,nsd002st003,prt002st003 prt001st003
GUI_MGMT_SERVERS      prt002st003,prt001st003
GUI_SERVERS           nsd001st003,prt003st003,nsd002st003,prt002st003 prt001st003
kafkaZookeeperServers  nsd001st003,prt003st003,nsd002st003
kafkaBrokerServers    prt003st003,prt002st003,prt001st003
```

Enable file system for auditing

You should select the file system that needs to be audited and enable it for file access auditing by using the **mmaudit** command.

In the following example, a file system called gpfs0 is enabled with default parameters:

```
# mmaudit all list
[I] File audit logging is disabled for all devices.

# mmaudit gpfs0 enable
[I] Successfully created File Audit Logging consumer node class
kafkaAuditConsumerServers
[I] Verifying MsgQueue nodes meet minimum local space requirements for File Audit
Logging to be enabled for device: gpfs0.
   Depending on cluster size, this may take some time.
[I] Successfully verified all configured MsgQueue nodes meet minimum local space
requirements for File Audit Logging to be enabled for device: gpfs0
[I] Successfully updated File Audit Logging configuration for device: gpfs0
[I] Successfully created File Audit Logging topic on the MsgQueue for device:
gpfs0
[I] Successfully enabled ACL access to the topic for producers and consumers for
device: gpfs0
[I] Successfully created/linked File Audit Logging audit fileset .audit_log with
link point /ibm/gpfs0/.audit_log
[I] Successfully enabled File Audit Logging consumer group to audit device: gpfs0
[I] Successfully created File Audit Logging policy partition(s) to audit device:
gpfs0
[I] Successfully created File Audit Logging consumer callbacks
[I] Successfully enabled File Audit Logging for device: gpfs0
```

The following example shows that file system gpfs0 is enabled for file audit logging. The audit logs reside on a fileset named .audit_log present on the gpfs0 file system itself (which is the default setting) with a retention of 365 days.

```
# mmaudit all list
Audit    Cluster                                       Fileset   Fileset
Retention
Device   ID                                             Device    Name
(Days)
---------------------------------------------------------------------------------
-------
gpfs0     11316316357699068975                     gpfs0     .audit_log
365
```

Note: From IBM Spectrum Scale 5.0.1, SASL SCRAM is used as the file audit logging authentication mechanism for improved security. The GUI has also been enhanced to monitor file auditing.

For more information about installation and deployment, see IBM Knowledge Center:

- https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1ins_enableauditlogging.htm#concept_tp4_j2m_j1b
- https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1ins_adlginstall.htm#adlginstall
- https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1ins_adlgreqlim.htm

7.2 Audit logging for cluster configuration changes

For problem determination and in auditing changes to the cluster configuration, audit messages can be sent to syslog or to the GPFS log whenever an IBM Spectrum Scale command changes the configuration of the cluster.

The facility is controlled by the **commandAudit** attribute of the **mmchconfig** command.

If audit logs are enabled, the GUI receives the updates on configuration changes that you make through the CLI, and updates its configuration cache to reflect the changes in the GUI. You can also disable audit logging by using the **mmchconfig** command. If the audit logs are disabled, the configuration changes made through the CLI are not reflected immediately in the GUI. For more information about audit messages for cluster configuration changes, see IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.0/com.ibm.spectrum.scale.v5r00.doc/bl1pdg_syslog.htm

7.2.1 Message format

Audit messages are sent to syslog with an identity of mmfs, a facility code of user, and a severity level of informational. Messages about IBM Spectrum Scale commands that are entered at the command line have the following format:

```
CLI user_name user_name [AUDIT_TYPE1,AUDIT_TYPE2] 'command' RC=return_code
```

Where:

CLI The source of the command. Indicates that the command was entered from the command line.

user_name user_name The name of the user who entered the command, such as root. The same name appears twice.

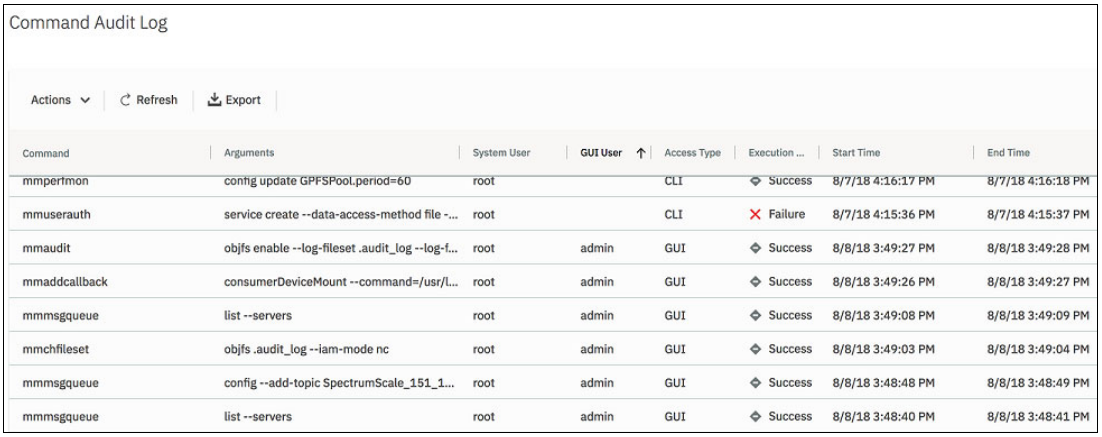
AUDIT_TYPE1 The point in the command when the message was sent to syslog. It is always EXIT.

AUDIT_TYPE2 The action that was taken by the command. It is always CHANGE.

Command The text of the command.

return_code The return code of the IBM Spectrum Scale command.

The format of entries that are generated by the GUI is similar, but includes the name of the user who logged in to the GUI, as shown in the Figure 7-4.



Command	Arguments	System User	GUI User ↑	Access Type	Execution ...	Start Time	End Time
mmperfmon	config update GPFSPool.period=60	root		CLI	◆ Success	8/7/18 4:16:17 PM	8/7/18 4:16:18 PM
mmuserauth	service create --data-access-method file -...	root		CLI	✕ Failure	8/7/18 4:15:36 PM	8/7/18 4:15:37 PM
mmaudit	objfs enable --log-fileset .audit_log --log-f...	root	admin	GUI	◆ Success	8/8/18 3:49:27 PM	8/8/18 3:49:28 PM
mmaddcallback	consumerDeviceMount --command=/usr/l...	root	admin	GUI	◆ Success	8/8/18 3:49:26 PM	8/8/18 3:49:27 PM
mmmsgqueue	list --servers	root	admin	GUI	◆ Success	8/8/18 3:49:08 PM	8/8/18 3:49:09 PM
mmchfileset	objfs .audit_log --iam-mode nc	root	admin	GUI	◆ Success	8/8/18 3:49:03 PM	8/8/18 3:49:04 PM
mmmsgqueue	config --add-topic SpectrumScale_151_1...	root	admin	GUI	◆ Success	8/8/18 3:48:48 PM	8/8/18 3:48:49 PM
mmmsgqueue	list --servers	root	admin	GUI	◆ Success	8/8/18 3:48:40 PM	8/8/18 3:48:41 PM

Figure 7-4 Command Audit Log

7.3 References

These websites are useful for further research:

- IBM Spectrum Scale documentation for file audit logging:

  https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1ins_quickrefadlg.htm

- IBM Spectrum Scale Audit Logging With Varonis DatAdvantage (For 4.2.x version of IBM Spectrum Scale only):

  https://ibm.biz/BdYGbT

- Varonis DatAdvantage:

  https://www.varonis.com/products/datadvantage

- IBM Spectrum Scale documentation: Audit messages for cluster configuration changes:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1pdg_syslog.htm

8

Hadoop security

Much of the initial focus of Hadoop development (circa 2006) was concentrated on the ability to store and process massive amounts of data. In the early days of Hadoop implementations and deployments, security was not a primary focus. Security was limited to a requirement that the Hadoop cluster and users accessing it are a part of a trusted network. As Hadoop adoption has grown in enterprises to run mission-critical workloads, the need for advanced security features such as strong authentication, authorization, and data security has become a necessity.

Businesses want to use typical enterprise security tools, such as auditing, encryption, and secure erase, on their Hadoop deployments. Since the adoption of Kerberos in 2010, Hadoop has been adding enterprise security features rapidly, as companies are entrusting sensitive data (such as healthcare records and financial information) to their Hadoop platforms.

This chapter includes the following sections:

- An introduction to Hadoop support in IBM Spectrum Scale: HDFS Transparency
- Kerberos
- Authentication
- Authorization
- Auditing
- Securing REST access
- Data protection
- Securing the Hadoop distribution components
- References

8.1 An introduction to Hadoop support in IBM Spectrum Scale: HDFS Transparency

IBM Spectrum Scale provides integration with the Hadoop framework through its second-generation connector called *HDFS Transparency*. Unlike the first-generation Hadoop connector, which replaced Hadoop Distributed File System (HDFS) as the file system in a Hadoop cluster, the HDFS Transparency integrates with HDFS.

In this approach, the HDFS client is used as-is and the HDFS Transparency connector implements the RPCs that the HDFS client needs to interact with the **NameNode** and **DataNode** services. The use of the native HDFS client, along with a **NameNode** service, ensures that most Hadoop applications and tools that interact with the Hadoop framework work without any changes.

HDFS Transparency integrates with and supports any Open Source Apache Hadoop compatible distribution. For more information, see the Hadoop on IBM Spectrum Scale wiki:

https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/General%20Parallel%20File%20System%20(GPFS)/page/Hadoop%20on%20Spectrum%20Scale

8.2 Kerberos

In 2009, Kerberos was selected to provide a common strong authentication service for Hadoop. Kerberos integrates with tools, such as Microsoft Active Directory (AD) and OpenLDAP. In Hadoop, both the component services and users can be authenticated through Kerberos. For more information, see the following topics:

- Kerberos in HortonWorks HDP 2.6.x

  https://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.6.4/bk_security/content/_kerberos_overview.html

- Hadoop in Secure Mode

  https://hadoop.apache.org

8.3 Authentication

By default, Hadoop supports a simple authentication mechanism. In this mechanism, the effective UID of the client is used to determine the identity of the user. No additional checks are performed, and the Hadoop servers trust the clients implicitly. Users are granted full access to all the data along with the administrative privileges on the Hadoop cluster.

This mechanism works when the physical and network access to the cluster is limited to a set of acceptable users. However, in production enterprise environments, the Hadoop cluster should be configured to use Kerberos as the authentication mechanism. Kerberos is an excellent authentication service, but has poor built-in support for authorization rules.

Deploying Kerberos with LDAP or as a part of AD provides a common identity for users who have authorization to run various Hadoop services and the users that access those services in a Hadoop cluster. This configuration has the added benefit of providing the mapping of a user to a group, which cannot be handled by Kerberos itself.

Hadoop is a collection of components that offer various services. These services and their protocols or interfaces support different, albeit a small number, of authentication methods. All services must be individually configured to enable Kerberos for authentication in a Hadoop cluster. Various Hadoop distributions provide mechanisms to enable Kerberos authentication for all the services in a guided manner.

For more information, see *HDFS Transparency Security Guide*:

`https://ibm.biz/BdYaTh`

8.4 Authorization

This section covers the various authorization methods that are available:

- HDFS
- Authorization for other Hadoop services

8.4.1 HDFS

HDFS implements an authorization mechanism similar to POSIX-compatible file systems. Permissions are managed in three separate scopes: *User*, *group*, and *others*. Files and directories are owned by a user, and the owner determines the user scope. Files and directories are assigned a group, and all members of the group determine the group scope. Users who are not the owner and do not belong to the group that is assigned to that file or directory comprise the others scope.

Each scope has a distinct set of permissions, such as `read`, `write`, and `execute`. These permissions can be granted to each scope independently. The `execute` permission is ineffective for files as they cannot run in HDFS. For directories, the `execute` permission gives access to the content and details of all the files in that directory.

Since Hadoop 2.4, HDFS added support for extended ACLs, which support additional permissions other than the basic POSIX ones. ACLs must be enabled in `hdfs-site.xml` and later added to files and directories, as by default they do not have any ACLs.

Regardless of the regular permissions or ACLs, the user that the **`NameNode`** service runs as (usually **`hdfs`**) is the equivalent of the `root` superuser on a Linux or UNIX system. This user can read, write, delete, and access any file or directory. Similarly, any member of the group that is defined as the `dfs.permissions.supergroup` (usually it is `hdfs` in HortonWorks HDP) has superuser privileges.

The IBM Spectrum Scale HDFS Transparency connector follows the same semantics for authorization as the native HDFS service. The configuration settings for HDFS, including security policies, are inherited by the HDFS Transparency connector, which are propagated to the native underlying IBM Spectrum Scale file system and enforced.

8.4.2 Authorization for other Hadoop services

In addition to HDFS permissions, various Hadoop services also support authorization. This service-level authorization is used to control user or group access to various components that might consume resources, such as memory and CPU. Configuration of these SLAs for various other Hadoop components, such as YARN, MapReduce, ZooKeeper, Hbase, and Hive, is beyond the scope of this chapter, but is a necessary and pertinent part of the overall Hadoop security model.

Enforcing a common security policy across all components is a complex task for Hadoop administrators. A central command-and-control mechanism is needed to implement common security policies across all of the Hadoop components in a uniform manner. This requirement has given rise to new Apache projects, such as Sentry and Ranger.

Although they vary in the particulars, both Sentry and Ranger aim to provide a central location for managing all security-related tasks, including role-based or attribute-based access control, fine-grained authorization, authentication, auditing, and data protection. For more information, see 8.9, “References” on page 60.

8.5 Auditing

Auditing is a mechanism by which a system can track the activities of the users and services interacting within the system. Most Hadoop components can be configured to passively log all activity. Generally, Key services (such as HDFS, MapReduce, YARN, Hive, HBase, Sentry, and Ranger) are configured to enable audit logging.

For example, HDFS can be configured to log all general user and all service-level authorization activity. Additionally, Ranger and non-Hadoop tools such as IBM Security Guardium® Data Activity Monitor can be used to actively monitor the system and provide alerts for targeted events.

For more information, see the *Security* topic:

https://www.ibm.com/us-en/products/category/technology/security

Auditing is also essential to achieve and maintain compliance in enterprise systems that contain sensitive data, such as medical records and financial information.

In a distributed system such as Hadoop, various components generate audit logs on each server in the cluster. Use log aggregation tools, such as syslog, to collect all relevant information in a central place and maintain a historical view over a set time. This configuration can aid in compliance and during forensic analysis after a breach is detected.

8.6 Securing REST access

Many of the Hadoop components provide REST APIs for interacting with the respective services. Configuring, managing, and logging the interaction of the REST APIs with the various services can be a challenge. Apache Knox Gateway provides a single access point for all REST interactions with the Hadoop cluster. Knox integrates with popular enterprise identity management services, and provides a single point of control, management, monitoring, and auditing of REST access to the Hadoop cluster.

8.7 Data protection

The following types of data protection are available:

- Data at rest
- Data in motion
- Secure Erase

8.7.1 Data at rest

IBM Spectrum Scale provides secure data at rest by offering a data encryption solution. Although the IBM Spectrum Scale encryption solution is not integrated with the native HDFS encryption solution, it provides a common enterprise-grade encryption solution for Hadoop and any other applications that use IBM Spectrum Scale as the underlying file system. By configuring Hadoop components such as MapReduce2 and Impala, or components such as BigSQL, to store intermediate data in directories on the IBM Spectrum Scale file system, encryption can be attained for temporary data for added protection.

Encryption is available with IBM Spectrum Scale Advanced Edition or Data Management Edition. For more information, see Chapter 2, “Secure data at rest” on page 7.

8.7.2 Data in motion

Hadoop uses three methods of communication over the network: RPC, TCP/IP, and HTTP. Each of these methods use a different data-in-motion encryption method.

Hadoop RPC encryption

Hadoop RPC encryption uses the Java SASL implementation, which supports the `auth`, `auth-int`, and `auth-conf` modes. Hadoop RPC encryption is enabled by using the `hadoop.rpc.protection` configuration parameter in `core-site.xml`.

HDFS data transfer protocol encryption

HDFS data transfer protocol uses direct TCP/IP sockets to transfer data between DataNodes, or between DataNodes and clients. Data transfer encryption can be enabled by setting the `dfs.encrypt.data.transfer` parameter in `hdfs-site.xml` to `true`. The Hadoop RPC protocol is used to exchange the encryption key for use in the encrypted data transfer, so HDFS RPC encryption should also be enabled.

Hadoop HTTP encryption

HTTP data in transit can be encrypted using HTTPS, which is an add-on of an SSL/TLS implementation on top of standard HTTP. Many Hadoop components support HTTPS, but there is no single configuration setting to enable HTTPS use across all the services, and each component must be individually configured to use HTTPS. For more information about how to enable HTTPS for each component, see the component-specific documentation (see 8.8, “Securing the Hadoop distribution components” on page 60).

8.7.3 Secure Erase

If encryption is enabled in IBM Spectrum Scale, then secure deletion of a set of files can be attained by deleting the file normally and removing the master encryption key (MEK) for those files. For more information, see Chapter 2, “Secure data at rest” on page 7.

8.8 Securing the Hadoop distribution components

Each Hadoop distribution includes a different set and version of components.

Distributions, such as Hortonworks and Cloudera, package a different set and version of components. You are also free to create a Hadoop stack by downloading the relevant packages from Apache. An enterprise adopting Hadoop must understand all of the components being deployed, and reconcile the security capabilities of these components with the prescribed enterprise security posture. For more information about how to secure HortonWorks HDP components, see the following resources:

- Hortonworks Data Platform Security

  https://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.6.5/bk_security/content/ch_hdp-security-guide-overview.html

8.9 References

The following websites are useful for further research:

- Hadoop on IBM Spectrum Scale wiki:

  https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/General%20Parallel%20File%20System%20(GPFS)/page/Hadoop%20on%20Spectrum%20Scale

- Kerberos in IBM Open Platform with Apache Hadoop 4.1:

  http://www.ibm.com/support/knowledgecenter/en/SSPT3X_4.1.0/com.ibm.swg.im.infosphere.biginsights.admin.doc/doc/admin_kerb_container.html

- Kerberos in IBM Open Platform with Apache Spark and Apache Hadoop:

  http://www.ibm.com/support/knowledgecenter/en/SSPT3X_4.2.0/com.ibm.swg.im.infosphere.biginsights.admin.doc/doc/admin_kerb_container.html

- Hortonworks Data Platform Security: ACLs on HDFS:

  https://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.6.4/bk_security/content/ch_acls_hdfs.html

9

Security for transparent cloud tiering

Transparent cloud tiering (TCT) is available as an add-on feature in IBM Spectrum Scale (Advanced Edition and Data Management Edition). It provides a native cloud storage tier as an external storage pool. TCT uses the existing information lifecycle management (ILM) policy infrastructure in IBM Spectrum Scale that enables administrators to define ILM policies to tier the data to cloud storage. The cloud storage tier should be used to store cooler (infrequently accessed) data from the IBM Spectrum Scale file system.

TCT enables tiering of file data from the IBM Spectrum Scale file system to IBM Cloud™ Object Storage to achieve storage efficiency and cost reduction. It frees primary storage by moving data to a low-cost object tier. A storage administrator can place data on IBM Cloud Object Storage, IBM SoftLayer® object storage, or Amazon S3 with the same ease with which you choose a local disk array.

Figure 9-1 shows a TCT configuration example.

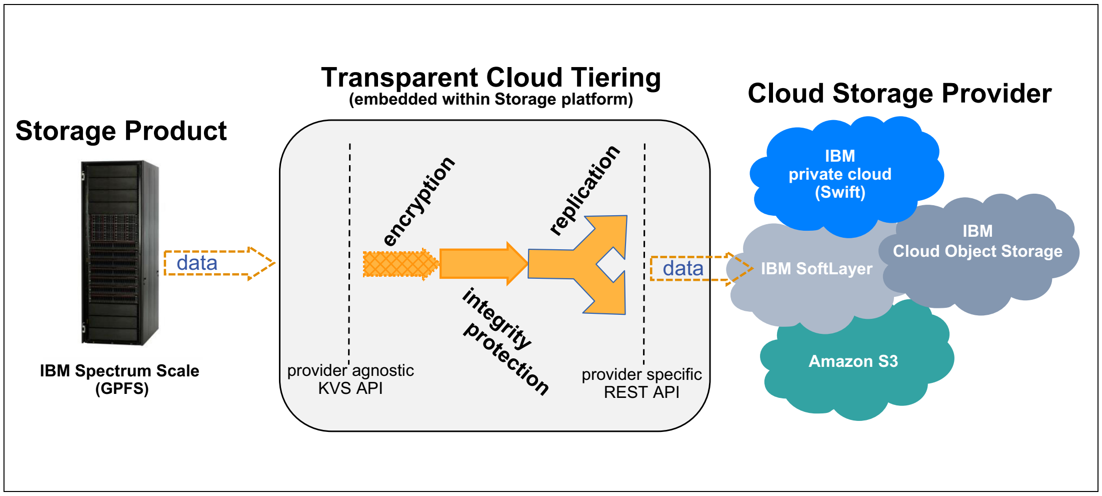

Storage Product
IBM Spectrum Scale (GPFS)
data
Transparent Cloud Tiering
(embedded within Storage platform)
encryption
replication
integrity protection
provider agnostic KVS API
provider specific REST API
data
Cloud Storage Provider
IBM private cloud (Swift)
IBM SoftLayer
IBM Cloud Object Storage
Amazon S3

Figure 9-1 TCT configuration

This chapter includes the following sections:

- Securing data in flight and at rest
- Securing the keys that are used to protect the data
- Configuring transparent cloud tiering with an external key manager: IBM Security Key Lifecycle Manager
- Configuring transparent cloud tiering with local key manager: Java Key Store
- TCT client-server communication security
- Security of TCT commands
- Data integrity protection
- Security considerations while configuring a cloud object storage
- References

9.1 Securing data in flight and at rest

TCT supports encryption of data on the client side. Data is encrypted before it is pushed to Cloud Object Storage (on-premises or off-premises). This ensures that the data is protected in motion and also at rest on the Object Storage layer as well. Data is protected in motion by using TLS.

For data encryption, TCT uses symmetric key encryption mechanism and adopts the AES algorithm with a 256-bit key length, which is a NIST-approved encryption mechanism.

TCT maintains the following two types of keys:

1. A key is used for encrypting file data during data movement to and from the cloud object storage. This key can be stored either in an external, shared IBM Security Key Lifecycle Manager (ISKLM) or within a local TCT managed JCE keystore.
2. A key is used internally by TCT to securely store configured cloud credentials. This key is always stored in a local TCT-managed JCE key store.

9.2 Securing the keys that are used to protect the data

TCT supports two types of Encryption Key Management Providers (IBM Security Key Lifecycle Manager and Java Key Store) to store the keys that are used for data encryption.

IBM Security Key Lifecycle Manager is an external key management provider. IBM Security Key Lifecycle Manager helps customers meet regulations and standards, such as the Payment Card Industry Data Security Standard (PCI DSS), Sarbanes-Oxley, and the Health Insurance Portability and Accountability Act (HIPAA), by providing centralized control and management of encryption keys.

TCT that is configured with IBM Security Key Lifecycle Manager is preferred for a production environment and is consistent with how IBM Spectrum Scale uses IBM Security Key Lifecycle Manager for data encryption.

By default, TCT supports Local Key Management by using Java Cryptography Extension Key Store. This configuration is ideal for test and non-production use cases.

Encryption of TCT objects is hierarchical: A master key encrypts a number of object keys, and an object key encrypts each of the objects. Object keys are chosen independently for each object.

While using local JCE key store for storing TCT encryption keys, it is important that an administrator periodically runs the **`mmcloudgateway service backupConfig`** command to back up the TCT configuration, including the key store. Data encrypted using keys stored in JCE key store cannot be recovered if the key store is lost.

For more information, see IBM Knowledge Center:

https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_backingupthecloudservicesconfiguration.htm

9.3 Configuring transparent cloud tiering with an external key manager: IBM Security Key Lifecycle Manager

For information on configuring ISKLM, see the following link:

https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_createkeymngr_multi.htm

After key manager for ISKLM is created, you can bind this to the container pair set. For more information about binding key manager to container pair set, see the following link:

https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_createcontainerpairset_multi.htm

For example, consider the following steps:

1. Create an SKLM Key manager named `vm1`:

   ```
   mmcloudgateway keymanager create --cloud-nodeclass  nodeclass
   --key-manager-name vm1  --key-manager-type RKM --sklm-hostname vm1 --sklm-port
   9080 --sklm-adminuser SKLMAdmin --sklm-groupname tct
   ```

   The previous command creates a device group, if it does not exist, and also generates a new encryption key under the specified device group.

2. Bind the container pair set using above key manager:

   ```
   mmcloudgateway containerpairset create --cloud-nodeclass tct
   --container-pair-set-name Containeretag5 --cloud-service-name csss5  --path
   /gpfs0/ --enc ENABLE --etag ENABLE --data-container test5 --meta-container
   testmeta5 --key-manager-name vm1
   ```

   Any data which will be migrated to the above data container will use encryption key created in Step1.

9.3.1 Rotating a key with IBM Security Key Lifecycle Manager

TCT supports rotation of encryption keys. The key rotation process creates an encryption key. All the new data that is going to be stored in IBM Cloud Object Storage is encrypted with the new encryption key. However, the old key is never deleted from the key store on a rotate-key operation, so that you can recall migrated data from the cloud. If the old key must be permanently removed from the key store for any reason, the administrator must manually recall all migrated data to avoid losing it permanently.

To perform a rotate key operation with IBM Security Key Lifecycle Manager, complete the following step:

```
mmcloudgateway keymanager rotate --cloud-nodeclass CloudNodeClass
--key-manager-name KeyManagerName
```

`rotate`

Rotates the existing key and creates a new SKLM key.

`--cloud-nodeclass CloudNodeClass`

Specifies the node class that was created by using the mmcrnodeclass command.

`--key-manager-name KeyManagerName`

Specifies the key manager name.

For example:

```
mmcloudgateway keymanager rotate --cloud-nodeclass nodeclass --key-manager-name
vm1
```

This example will generate a new SKLM encryption key and all the container pair sets using `key manager: vm1` will start using the new encryption key for any new data migration.

9.4 Configuring transparent cloud tiering with local key manager: Java Key Store

You can create the local key manager by using the following command:

```
mmcloudgateway keymanager create --cloud-nodeclass CloudNodeClass
--key-manager-name KeyManagerName --key-manager-type LKM --alias alias
```

`create`

Uses an SKLM key manager with Cloud services.

`--cloud-nodeclass CloudNodeClass`

Specifies the node class that is associated with the Cloud services.

`--key-manager-name KeyManagerName`

Specifies the key manager name.

`--key-manager-type`

Specifies the type of the key manager. If it is a remote key manager, specify `RKM`, and if it is a local key manager, specify `LKM`.

`--alias Alias`

Specifies the alias name of the local key manager encryption key.

When the key manager for a local key manager has been created, you can bind it to the container pair set. For more information about how to bind a key manager to a container pair set, see the following link:

https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_createcontainerpairset_multi.htm

For example, consider the following steps:

1. Create a Local Key manager named `localkm1`:

   ```
   mmcloudgateway keymanager create --cloud-nodeclass  nodeclass
   --key-manager-name localkm1--key-manager-type LKM –alias localkey1
   ```

   The above command creates an encryption key in local key store.

2. Bind the container pair set using above key manager:

   ```
   mmcloudgateway containerpairset create --cloud-nodeclass tct
   --container-pair-set-name Containeretag5 --cloud-service-name csss5  --path
   /gpfs0/ --enc ENABLE --etag ENABLE --data-container test5 --meta-container
   testmeta5 --key-manager-name localkm1
   ```

Any data which will be migrated to the above data container will use the encryption key created in Step1.

9.4.1 Rotating an encryption key with Local Key Manager

TCT supports rotation of encryption keys. The key rotation process creates an encryption key. All the new data that is going to be stored in IBM Cloud Object Storage is encrypted with the new encryption key. However, the old key is never deleted from the key store on a rotate-key operation, so that you can recall migrated data from the cloud. If the old key must be permanently removed from the key store for any reason, the administrator must manually recall all migrated data to avoid losing it permanently. From a security perspective, you should back up the CCR data regularly and store it in a safe location.

To perform a rotate key operation with Local Key manager, complete the following step:

```
mmcloudgateway containerpairset update --cloud-nodeclass tct
--container-pair-set-name Containeretag5 –active-key lkm2
```

Any data that will be migrated to this data container will use the encryption key associated with alias `lkm2`.

9.5 TCT client-server communication security

TCT commands can be invoked from the TCT server as well as TCT client nodes. All of the commands are sent to TCT Java service (back-end) for processing. TCT Java service implements an HMAC-based authentication mechanism to ensure that it is receiving requests only from valid nodes within the cluster. This authentication mechanism leverages the existing IBM Spectrum Scale certificate available for each node.

Along with the HMAC, every request also carries a time stamp indicating request initiation time to avoid any replay attack. Currently, all communication between a TCT client node and a server node is secured in a similar way.

9.6 Security of TCT commands

Only root users can start transparent cloud tiering CLI commands, which prevents attempts to restore or use another user’s file under a user’s own permissions, thus using TCT to bypass file system’s ACLs.

9.7 Data integrity protection

TCT enforces integrity protection of the outsourced data by using the *etag* that is received from the cloud provider by using a hash function that is specified by the cloud provider. Integrity protection relies on trusting the cloud storage provider. The TLS protocol is used when communicating with the cloud.

For example:

During a `Put` operation, the Transparent Cloud Tiering etag Module computes the etag of each blob using the cloud provider-based algorithm. At the end of the put blob operation, the etag received from the cloud provider and the Transparent Cloud Tiering etag Module computed etag should be matched, which ensures data integrity.

During a `Get` Operation, the Cloud provider etag is fetched using cloud stored metadata information and the Transparent Cloud Tiering etag Module computes the etag for each slice of the blob. At the end of the get blob operation, the etag received from the cloud provider and the Transparent Cloud Tiering etag Module computed etag should be matched, which ensures data integrity.

This etag-based integrity protection guarantees that the data on the wire is not tampered in transit.

9.8 Security considerations while configuring a cloud object storage

During the configuration of a cloud object storage account by using TCT, the following security considerations are handled:

- Account credentials, such as access key (for Amazon S3 and IBM Cloud Object Storage) or password (for SoftLayer Object Storage) are validated.
- TCT validates whether an Account user has the correct permission to create a storage container under the object storage provider.

9.9 References

The following websites are useful for further research:

- Preparing for encryption:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_encryption_prep.htm

- Creating a cloud storage account:

  https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_createacloudaccount_multi.htm

10

Security for OpenStack drivers

OpenStack is an open source set of software components that are used to build a private, public, or hybrid cloud environment. Within these components, the OpenStack platform consists of a number of cloud storage services, which can be deployed on IBM Spectrum Scale storage.

This chapter includes these sections:

- OpenStack components
- OpenStack components and IBM Spectrum Scale security
- References

10.1 OpenStack components

OpenStack has the following available components:

Cinder This component provides virtualized block storage (in the form of volumes) to virtual machines (VMs). IBM Spectrum Scale provides an OpenStack Driver for Cinder service, which is available with OpenStack distributions.

Glance This component provides the capability to manage VM images. Glance can be configured to use IBM Spectrum Scale for hosting VM images.

Swift This component provides object storage to a user or application that requires access to data through a RESTful API. Swift is part of an IBM Spectrum Scale protocol node that can provide unified file and object access to the data.

Manila This project provides shared file system access to any client (virtual or physical). IBM Spectrum Scale has a Manila driver to provide the shared file system by using NFS.

10.2 OpenStack components and IBM Spectrum Scale security

From an overall security perspective, the IBM Spectrum Scale storage drivers for OpenStack components rely mostly on OpenStack security features because they serve as back-end drivers. Detailed information about OpenStack security is available at the following website:

http://docs.openstack.org/security-guide

IBM Spectrum Scale provides additional security features that are available beyond those mentioned with the components drivers:

- Secure Data at Rest: IBM Spectrum Scale provides file encryption that ensures secure storage of data at rest, which is covered in detail in Chapter 2, “Secure data at rest” on page 7. For Cinder and Glance, the services administrator can choose to have an encrypted fileset to store volume files and VM images. The IBM Spectrum Scale Cinder driver can be configured to use multiple back ends that can be based on either encrypted filesets or non-encrypted filesets.

  Based on business needs, administrators can define different cinder volume types, where one volume type can map to encrypted back-end storage (an encrypted fileset) and the other volume type can map to unencrypted back-end storage (non-encrypted back-end storage). Using these volume types, administrators can then create volumes on either encrypted filesets or non-encrypted filesets as required.

  Similarly, for Glance services, you can choose to have an encrypted fileset to store VM images. IBM Spectrum Scale protocols such as Object and NFS (used by Manila) can be deployed on encrypted filesets, which helps enhance the security of data being deployed on OpenStack Deployment.

- Secure Data in Flight: Volumes that are hosted by the IBM Spectrum Scale Cinder driver can be accessed by using NFS and by a native IBM Spectrum Scale client. When access is done by using NFS, you can optionally use Kerberized NFS that is supported by IBM Spectrum Scale protocol nodes to secure the volume access. This configuration ensures that the volumes that are hosted by the Cinder driver are accessed securely.

For the shared file system service Manila, the IBM Spectrum Scale driver supports NFS through kNFS, and NFS Ganesha through IBM Spectrum Scale protocol nodes. You can use Kerberized NFS to ensure that the data that is shared through IBM Spectrum Scale Manila driver is secure over the wire. Additionally, the IBM Spectrum Scale Manila driver can configure access rules on the NFS shares based on IP addresses, which can be used to securely share data across OpenStack instances.

Details of secure data in flight by using IBM Spectrum Scale protocols, such as NFS (used by Manila service) and Object, and the authentication that is supported by these protocols, are covered in their respective chapters.

For more information about OpenStack deployment over IBM Spectrum Scale, see *IBM Spectrum Scale in an OpenStack Environment*, REDP-5331.

For more information about IBM Spectrum Scale in an OpenStack cloud deployment, see IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1ins_openstackusecase.htm

10.3 References

For more information about the topics that are described in this chapter, see the following resources:

- *IBM Spectrum Scale in an OpenStack Environment*, REDP-5331

  http://www.redbooks.ibm.com/abstracts/redp5331.html
- IBM Spectrum Scale in an OpenStack cloud deployment: IBM Spectrum Scale documentation:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1ins_openstackusecase.htm
- OpenStack Security Guide:

http://docs.openstack.org/security-guide/

11

Security for AFM

Active file management (AFM) is a scalable, high-performance, file system caching layer integrated with the GPFS cluster file system. AFM allows you to create associations from a local GPFS cluster to a remote cluster or storage, and to define the location and flow of file data to automate the management of the data. AFM uses a home-and-cache model in which a single home cluster provides the primary storage of data, and exported data is cached in a local GPFS file system.

AFM-based Async Disaster Recovery is an AFM-based fileset level replication disaster recovery capability to augment the overall business recovery solution.

This chapter includes the following security considerations for AFM:

- AFM and Authentication/ID Mapping
- AFM and secure data in transit
- References

11.1 AFM and Authentication/ID Mapping

When data is transferred between sites, you must ensure that the ownership of the file (UID/GID) is consistent between the sites. AFM transfers data between sites but it does not replicate Authentication and User ID configuration setup between the AFM sites. It is the administrator's responsibility to configure similar authentication and ID mapping across the sites. The AFM layer ensures that the metadata consisting of file ownership and ACL is appropriately replicated.

11.2 AFM and secure data in transit

The AFM layer does not encrypt data when it moves from one site to another. To ensure that the data transferred through AFM is secured, either of the following security measures must be applied:

1. The administrator can configure any external security measures, such as VPN, between the two sites that AFM has been configured upon.
2. From IBM Spectrum Scale 5.0.1 onwards, AFM has the added support of mounting Kerberos-enabled exports for an NFS backend. Figure 11-1 shows the different levels of security, the administrator can deploy for AFM when Kerberized NFS is used as its underlying protocol.

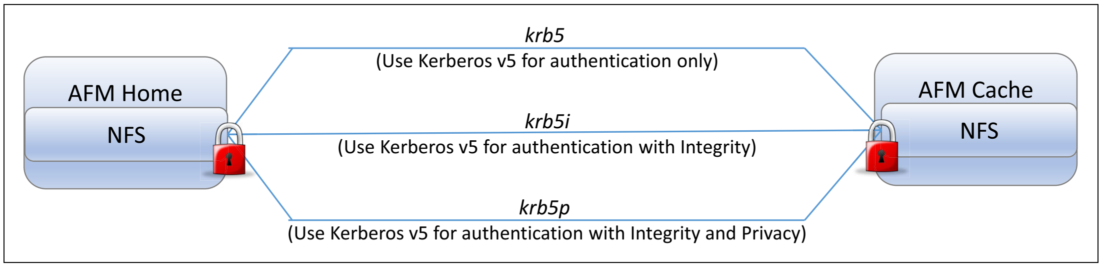

krb5
(Use Kerberos v5 for authentication only)
AFM Home
NFS
krb5i
(Use Kerberos v5 for authentication with Integrity)
AFM Cache
NFS
krb5p
(Use Kerberos v5 for authentication with Integrity and Privacy)

Figure 11-1 AFM over Kerberized NFS

For remote cluster mount (NSD) backend, AFM leverages the supported encryption features for remote file systems. Figure 11-2 shows the AFM security when NSD is used as its underlying protocol.

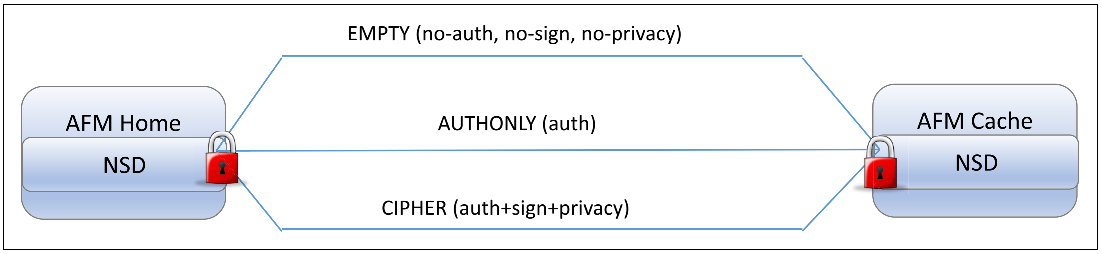

EMPTY (no-auth, no-sign, no-privacy)
AFM Home
NSD
AUTHONLY (auth)
AFM Cache
NSD
CIPHER (auth+sign+privacy)

Figure 11-2 AFM over NSD protocol

11.3 References

The following websites are useful for more detailed research:

- Quick reference section: Active File Management (AFM)

  https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/b1lins_quickreference_afm.htm

- Quick reference section: AFM-based Disaster Recovery

  https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/b1lins_quickreference_afmbaseddr.htm

12

Firewall recommendations

IBM Spectrum Scale can function within a wide range of network environments, from open to highly secure. When configuring firewalls on networks that support IBM Spectrum Scale, it is important to understand the network port usage of each function. Many functions require access to specific ports, and other functions must be told which ports to use.

Open environments can choose to disable firewalls completely, allowing all ports to communicate from within and externally to the IBM Spectrum Scale cluster. Highly secure environments might lock down each OS participating both within the IBM Spectrum Scale cluster and those accessing the cluster from an external entry point, opening only the ports that are necessary for active functions.

The default firewall state (running or inactive), default open/closed ports, default firewall zones, and default port/service mappings differ among the various operating systems that are supported by IBM Spectrum Scale. This chapter outlines various functions within IBM Spectrum Scale and details their networking port requirements. For more information see Appendix A, “Examples of how to open firewall ports” on page 89.

This chapter includes the following sections:

- Types of networks
- IBM Spectrum Scale installation and basic cluster operation
- GUI
- Performance Monitoring tools
- Transparent cloud tiering
- Cluster Export Services
- File audit logging
- Active File Management
- IBM Spectrum Scale remote mounting of file systems
- IBM Spectrum Protect connectivity by using mmbackup and HSM
- IBM Spectrum Archive connectivity
- IBM Spectrum Control connectivity
- Key server ports
- References

12.1 Types of networks

Two networks are referenced throughout this chapter:

- The internal intra-cluster network

  This is a network that is used within an IBM Spectrum Scale cluster for node-to-node communication. It is critical for cluster operations, such as bringing nodes online and making them active, mounting one or more shared file systems, and passing data between all nodes back to the NSD servers. Day-to-day cluster administration tasks and data exchange occur over this network.

- Cluster Export Services (CES) network

  This is a network that is used when choosing to externally present IBM Spectrum Scale nodes and services. This network can be the same as the intra-cluster network, but often is separate to reduce accessibility down to a specific set of services. Protocols, such as SMB, NFS, and Object, can be used to externally access data that is shared by an IBM Spectrum Scale cluster through one of these external client-facing networks.

  GUI management of a cluster can be surfaced to this network, thus enabling remote cluster management. Cross-site replication through AFM, Backup, and tiering of data to an IBM Spectrum Protect™ server are functions that require access to one or more networks.

Figure 12-1 shows an intra-cluster network.

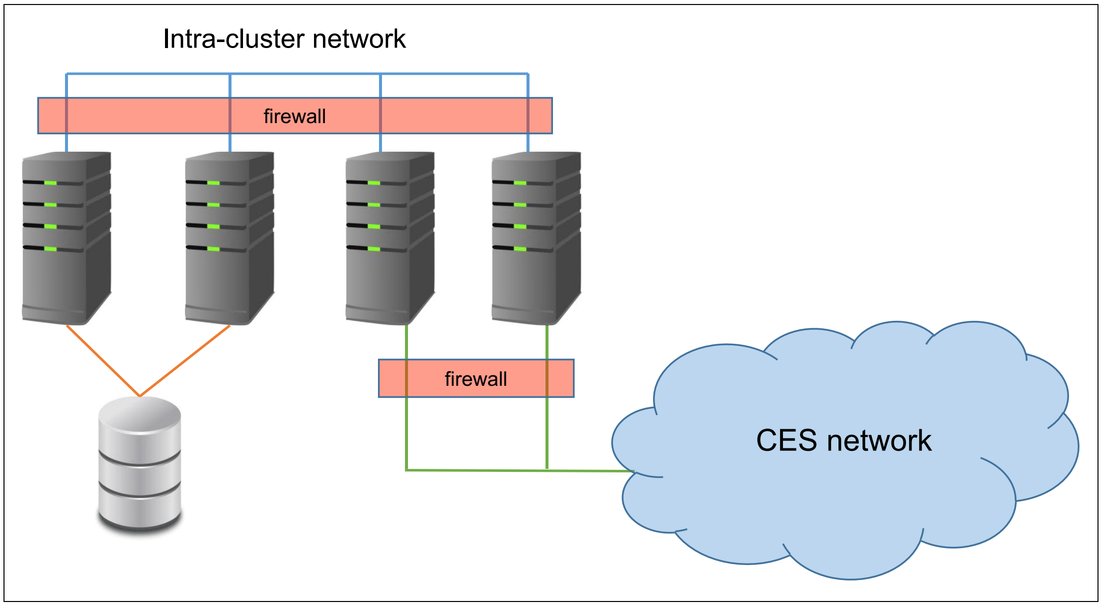


Figure 12-1 Diagram of an intra-cluster network

12.2 IBM Spectrum Scale installation and basic cluster operation

The IBM Spectrum Scale Installation Toolkit can be used for these tasks:

- Installation of a basic IBM Spectrum Scale cluster
- Creation of NSDs and file systems
- Setup of performance monitoring tools, installation and activation of the management GUI, and CES

The Installation Toolkit uses the ports that are listed in Table 12-1.

Table 12-1 Installation Toolkit port requirements

Port number	Protocol	Service name	Components that are involved in communication
8889	TCP	Chef	Intra-cluster and installer server
10080	HTTP	Chef	Python Web framework
123	UDP	NTP	Intra-cluster or external depending on NTP server location

Chef is the underlying technology that drives the Installation Toolkit. During installation, a Chef server is started on the installation server, and repositories are created to house the various IBM Spectrum Scale components. Each node that is installed by the Installation Toolkit must be able to establish a connection to the repository and the Chef server itself. Chef uses ports 8889 and 10080 to communicate with the nodes being installed.

NTP is not required for most IBM Spectrum Scale services, but is highly preferred. NTP is required for protocol nodes.

When the installer node is not part of the cluster, passwordless SSH needs to be set up from the installer node to all other nodes in the cluster. For more information on setting up the passwordless SSH, see IBM Knowledge Center:

https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1ins_preparingtousetoolkit.htm

Table 12-2 lists the port requirements for basic GPFS cluster operation.

Table 12-2 Basic GPFS cluster operation port requirements

Port number	Protocol	Service name	Components that are involved in communication
1191	TCP	GPFS	Intra-cluster
User-selected range	TCP	GPFS Ephemeral Port Range	Intra-cluster
22	TCP	SSH	Intra-cluster and administrative access to the cluster nodes

If you are using SSH for remote shell and remote copy, port 22 is used for command execution and general administrative access.

The primary GPFS daemons (**mmfsd** and **mmsdrserv**), by default, listen on port 1191. This port is essential for basic cluster operation. The port can be changed by using the **tscTcpPort** and **mmsdrservPort** configuration variables:

```
mmchconfig mmsdrservPort=PortNumber
mmchconfig tscTcpPort=PortNumber
```

The ephemeral port range of the underlying OS is used when IBM Spectrum Scale creates extra sockets to exchange data among nodes. This process occurs while running certain commands and is dynamic, based on the point-in-time needs of the command and other concurrent cluster activities. A user can define an ephemeral port range by using the **`tscCmdPortRange`** configuration variable:

```
mmchconfig tscCmdPortRange=LowNumber-HighNumber
```

If the Installation Toolkit is used, the ephemeral port range is automatically set to 60000 - 61000. Firewall ports can be opened according to the ephemeral port range that is defined to IBM Spectrum Scale.

A sign of an improperly configured ephemeral port range is a hang with commands such as **`mmlsmgr`** and **`mmcrfs`**.

12.3 GUI

The following two IBM Spectrum Scale GUIs are available:

- The Install GUI is available through the Installation Toolkit for a first-time cluster. The Install GUI is available from a web browser by using either the HTTP or the HTTPs protocol.
- The management GUI is used for day-to-day cluster activities, such as viewing overall cluster state, events, and administering filesets, snapshots, protocols, ILM, ACLs, and diagnostic data. From 4.2.3 release onwards, the required ports have changed because the GUI process is running as a non-root user. Privileged ports like 80 or 443 are not allowed for non-root processes.

  Therefore, the GUI web server which is running under the `scalemgmt` user listens on ports 47080 and 47443 instead. To ensure access to the GUI via ports 80 and 443, port forwarding is implemented at startup through different `iptables` rules. See Table 12-3 for the list of port numbers. For more information about the implications of using a non-root user, see IBM Knowledge Center:

  https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1ins_guiconfignonrootprivileges.htm

Note: REST API is accessible via the secure ports 443 and 47443 only.

Table 12-3 GUI port requirements

Port number	Protocol	Service name	Components that are involved in communication
9080	TCP	Install GUI (optional)	HTTP external network
9443	TCP	Install GUI (optional)	HTTPS external network
80	TCP	Management GUI < 4.2.3, IPTABLES redirect >= 4.2.3	HTTP external network
443	TCP	Management GUI < 4.2.3, IPTABLES redirect >= 4.2.3 REST API	HTTPS external network

Port number	Protocol	Service name	Components that are involved in communication
47080	TCP	Management GUI >= 4.2.3	HTTP external network
47443	TCP	Management GUI >= 4.2.3 REST API	HTTPS external network
4444	TCP	Management GUI CLI Server	localhost only
(See Table 12-4.)	TCP	Zimon Collector	Intra-cluster

> **Note:** All nodes in the cluster can access the GUI(s) on ports 80 and 443. These ports are used for CLI auditing, callbacks, and System Health events that are sent to the GUI(s).

If the Object GUI capabilities are enabled, the GUI will act as a client for the object stack. The GUI will need access to the object client and keystone ports on the CES nodes. See Table 12-8 for details on the object protocol.

12.4 Performance Monitoring tools

IBM Spectrum Scale Performance Monitoring tools consist of sensor and collector components. By default, sensors are installed and activated on all nodes, and collectors are installed and activated on GUI nodes. Sensors send data to the collectors by using TCP and the ports that are outlined in Table 12-4.

Table 12-4 Performance monitoring sensor and collector port requirements

Port number	Protocol	Service name	Components that are involved in communication
4739	TCP and UDP	Performance Monitoring Tool	Intra-cluster (used for GUI operation as well)
8123	TCP	Object Metric collection	Intra-cluster
8124	TCP	Object Metric collection	Intra-cluster
8125	TCP	Object Metric collection	Intra-cluster
8126	UDP	Object Metric collection	Intra-cluster
8127	TCP	Object Metric collection	Intra-cluster
9084	TCP	Performance Monitoring Tool	Any node that wants to query the database (used for GUI operation as well)
9085	TCP	Performance Monitoring Tool	Intra-cluster (used for GUI operation as well)
9094	TCP	Performance Monitoring Tool	Any node that wants to query the database (used for GUI operation as well)

Note:

- The 4739 port must be open when a collector is installed.
- The 9085 port must be open when there are two or more collectors.
- If the 9084 port is closed, then accessing the collector to debug or to connect external tools or even another instance of the GUI remotely is not possible, except from the node where the GUI and collector are installed.

12.5 Transparent cloud tiering

IBM Spectrum Scale transparent cloud tiering (TCT) enables the IBM Spectrum Scale cluster to interface directly with cloud storage types, such as Amazon, IBM Cloud Object Storage, and OpenStack. Configuration of TCT requires that the TCT nodes be able to talk to each other within the network. It also requires a connection to the external network, which depends on the provider of Object Storage.

Table 12-5 lists the port requirements for TCT.

Table 12-5 Transparent cloud tiering port requirements

Port number	Protocol	Service name	Components that are involved in communication
8085	TCP	TCT	Intra-cluster
(Object storage provider dependent)	TCP	TCT	TCT connection to object storage provider on the external network. Typically HTTPS (443) or HTTP (80)

The intra-cluster port that is used by TCT can be changed from 8085 to any other port by using the following command:

```
mmcloudgateway config
```

Contact the Object Storage provider to understand what ports are needed to communicate with the object storage. Although this connection typically occurs with HTTPS (443) or HTTP (80), it is highly dependent on the chosen provider.

12.6 Cluster Export Services

CES support four protocols: NFS, SMB, Object, and iSCSI. CES protocols require a floating pool of IPs to be defined. The IPs are allowed to float between designated protocol nodes and can actively move around in cases of failure, for example. CES IPs are automatically assigned and aliased to existing network adapters on protocol nodes during startup. Because these CES IPs present access to data that is shared by NFS, SMB, Object, and iSCSI, it is important to consider them when designing a firewall implementation.

Example 12-1 shows aliased CES IPs by using the **`ip addr`** command.

Example 12-1 Showing aliased CES IPs

```
eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq state UP qlen 1000
    link/ether 00:50:56:83:16:e5 brd ff:ff:ff:ff:ff:ff
```

```
inet 10.11.1.122/24 brd 10.11.1255 scope global eth1
       valid_lft forever preferred_lft forever
inet 10.11.1.5/24 brd 10.11.1.255 scope global secondary eth1:0
       valid_lft forever preferred_lft forever
inet 10.11.1.8/24 brd 10.11.1.255 scope global secondary eth1:1
       valid_lft forever preferred_lft forever
```

In Example 12-1, interface `eth1` pre-exists with an established route and the IP `10.11.1.122`. This interface is assigned and must be accessible before any CES configuration. After CES services are active, CES IPs are then automatically aliased to this base adapter, creating `eth1:0` and `eth1:1`. The floating CES IPs that are assigned to the aliases are `10.11.1.5` and `10.11.1.8`. Both CES IPs are allowed to move to other nodes during a failure. This automatic movement, when combined with the ability to manually move CES IPs, can cause a variance in the number of aliases and CES IPs among protocol nodes.

Firewall rules can reference Ethernet adapters, subnets, and individual IPs, but not adapter aliases. Due to this limitation, create inbound and outbound firewall rules based on the subnet, all CES IPs themselves, or the base adapter. These rules must account for all wanted CES protocols. The following sections provide the specific port requirements of NFS, SMB, and Object.

12.6.1 NFS file protocol (Cluster Export Services)

NFS is one of four protocols that are provided by IBM Spectrum Scale CES. NFS is typically used to give data access to a client on an external network. All IBM Spectrum Scale protocol nodes that run in NFS must have firewall ports open to allow connections from clients to all CES IP addresses. Likewise, the client accessing the IBM Spectrum Scale cluster must have firewall ports open to allow incoming and outgoing connections.

IBM Spectrum Scale Active File Management (AFM) uses NFSv3.

The NFS file protocol has the port requirements that are listed in Table 12-6.

Table 12-6 NFS file protocol port requirements

Port number	Protocol	Service name	Components that are involved in communication
2049	TCP and UDP	NFSv4 or NFSv3	NFS clients, AFM, and IBM Spectrum Scale protocol node
111	TCP and UDP	portmapper (required only by NFSv3)	NFS clients, AFM, and IBM Spectrum Scale protocol node
User-defined static port	TCP and UDP	NSM (required only by NFSv3)	NFS clients, AFM, and IBM Spectrum Scale protocol node
User-defined static port	TCP and UDP	MOUNT (required only by NFSv3)	NFS clients, AFM, and IBM Spectrum Scale protocol node
User-defined static port	TCP and UDP	nlm (required only by NFSv3)	NFS clients, AFM, and IBM Spectrum Scale protocol node
User-defined static port	TCP and UDP	rquota (NFSv3 or NFSv4)	NFS clients and IBM Spectrum Scale protocol node

NFSv3 uses dynamic ports for NLM, MOUNT, and NSM services. When an NFSv3 server is used with the firewall, these services must be configured with static ports.

The following preferred practices are applicable:

- Set static ports for MOUNT, NLM, and NSM services that are required by the NFSv3 server by using the **mmnfs configuration change** command. Allow TCP and UDP port 2049 to use the protocol node IPs, as shown in the following example:

```
mmnfs configuration change
MNT_PORT=32767:NLM_PORT=32769:RQUOTA_PORT=32768:STATD_PORT=32765
```

- Enable all external communications on TCP and UDP port 111 by using the protocol node IPs.
- Enable all external communications on the TCP and UDP port that is specified by the **mmnfs configuration change** command for MOUNT and NLM ports.
- Restart NFS after changing these parameters by using the following commands:

```
mmces service stop NFS -a
mmces service start NFS -a
```

- Use **rpcinfo -p** to query the protocol nodes after any port changes to verify that the proper ports are in use.
- Remount any existing clients, because a port change might have disrupted connections.

12.6.2 SMB file protocol (Cluster Export Services)

SMB is one of four protocols that are provided by IBM Spectrum Scale CES. All IBM Spectrum Scale protocol nodes that run SMB must have firewall ports open to allow connections from clients. Likewise, the client accessing the IBM Spectrum Scale cluster must have firewall ports open to allow incoming and outgoing connections.

Because SMB relies on the internal CTDB component for storing configuration information, all protocol nodes must have their CTDB port open on the internal cluster network.

The SMB file protocol has the following port requirements (Table 12-7).

Table 12-7 SMB file protocol port requirements

Port number	Protocol	Service name	Components that are involved in communication
445	TCP	Samba	SMB clients and IBM Spectrum Scale protocol node
4379	TCP	CTDB	Intra-cluster (between protocol nodes only)

12.6.3 Object protocol (Cluster Export Services)

Object is one of four protocols that are provided by IBM Spectrum Scale CES. All IBM Spectrum Scale protocol nodes that run Object must have firewall ports open to allow connections from clients. Likewise, the client accessing the IBM Spectrum Scale cluster must have firewall ports open to allow incoming and outgoing connections.

Many Object operations require communication within the Object node itself. These situations are listed in Table 12-8 on page 83Table 12-8 on page 83 with the ports that are necessary for local host accessibility.

In order for authentication with Object by way of Keystone to work, it is necessary for all Object clients to have access to the Keystone ports.

Table 12-8 Object protocol port requirements

Port number	Protocol	Service name	Components that are involved in communication
8080	TCP	Object Storage Proxy	Object clients and IBM Spectrum Scale protocol node
6200	TCP	Object Storage (local object server)	Local host
6201	TCP	Object Storage (local container server)	Local host
6202	TCP	Object Storage (local account server)	Local host
6203	TCP	Object Storage (object server for unified file and object access)	Local host
11211	TCP and UDP	Memcached	Local host
5000	TCP	Keystone Public	Authentication clients and object clients
35357	TCP	Keystone Internal/Admin	Authentication clients and object clients and Keystone administrator
5431	TCP and UDP	postgresql-obj	Intra-cluster (between protocol nodes only)

12.6.4 iSCSI protocol (Cluster Export Services)

The iSCSI protocol is one of the four protocols that are provided by IBM Spectrum Scale CES. This protocol, which is supported by the BLOCK service, provides an iSCSI target service for remotely booting nodes. All IBM Spectrum Scale protocol nodes that run the BLOCK service must have firewall ports open to allow connections from clients.

Table 12-9 shows the required port information.

Table 12-9 Port information for IBM Spectrum Scale protocol node with iSCSI target service

Port number	Protocol	Service name	Components that are involved in communication
3260	TCP	BLOCK (iSCSI)	iSCSI client and IBM Spectrum Scale protocol node

This port needs to be opened only on the protocol nodes, and not on the clients themselves.

12.7 File audit logging

Table 12-10 shows the required port information.

Table 12-10 Port Information for File Audit Logging

Port number	Protocol	Service name	Components that are involved in communication
9092	TCP	IBM Spectrum Scale	File audit logging
9093	TCP	IBM Spectrum Scale	File audit logging
2181	TCP	IBM Spectrum Scale	File audit logging
2888 - 3888 (1000 ports)	TCP	IBM Spectrum Scale	File audit logging

12.8 Active File Management

IBM Spectrum Scale AFM allows one or more IBM Spectrum Scale clusters (or a non-IBM Spectrum Scale NFS source) to exchange file data.

AFM can be implemented by using the NFS protocol or NSD protocol:

- NFS AFM implementation: See Table 12-6 on page 81 for NFSv3 requirements.
- NSD AFM implementation: See Table 12-2 on page 77 for basic GPFS cluster operation port requirements.

12.9 IBM Spectrum Scale remote mounting of file systems

IBM Spectrum Scale clusters can access file systems of other IBM Spectrum Scale clusters by using remote mounts. This access can occur in two ways:

- All nodes in the IBM Spectrum Scale cluster that require access to another cluster’s file system must have a physical connection to the disks containing file system data. This is typically done by using SAN. Although outside the topic of firewalls, a SAN that is open to multiple clusters should also be subject to scrutiny from a security point of view.
- All nodes in the IBM Spectrum Scale cluster requiring access to another cluster’s file system must communicate to the NSD servers hosting the file system.

In both cases, all nodes in the cluster requiring access to another cluster’s file system must be able to open a TCP/IP connection to every node in the other cluster. For the basic GPFS cluster operation port requirements, see Table 12-2 on page 77.

Each cluster that is participating in a remote mount can be on the same network or a separate network from the host cluster. This configuration means, from a firewall standpoint, that the host cluster might need ports that are opened to a number of networks, depending on how many separate clusters are accessing the host.

12.10 IBM Spectrum Protect connectivity by using mmbackup and HSM

The **mmbackup** command is an IBM Spectrum Scale function that allows the backup of items, such as file systems and filesets, to an IBM Spectrum Protect server.

Hierarchical storage management (HSM) is used extensively with the policy engine to allow automatic storage tiering to an external disk or tape pool within an IBM Spectrum Protect server.

Both functions necessitate open communication between the nodes that are designated for use with **mmbackup** or HSM policies and the external IBM Spectrum Protect server. The port requirements in Table 12-11 can be viewed within the `dsm.sys` configuration file as well.

Table 12-11 IBM Spectrum Protect using HSM and mmbackup port requirements

Port number	Protocol	Service name	Components that are involved in communication
1500	TCP	TSM	TSM BA client communication with server

For more information and port requirements that are specific to the server end, see IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSGSG7/landing/welcome_ssgsg7.html

12.11 IBM Spectrum Archive connectivity

IBM Spectrum Archive™ software runs on a node or group of nodes within an IBM Spectrum Scale cluster, which means that each IBM Spectrum Archive node must allow communication with the rest of the cluster by using the same ports that are listed in Table 12-2 on page 77. Included ports are 1191, 22, and the ephemeral port range.

Communication between IBM Spectrum Scale and IBM Spectrum Archive happens within the same node. A couple of fixed ports and dynamic ports that are provided/resolved by **rpcbind** are used. Because the range of ports cannot currently be limited, the firewall services must be disabled.

IBM Spectrum Archive connects to tape drives through a SAN or direct connect. Although outside the topic of firewalls, the Fibre Channel connectivity that is used by IBM Spectrum Archive nodes should be reviewed from a security point of view.

12.12 IBM Spectrum Control connectivity

IBM Spectrum Control™ enables monitoring of multiple products within a data center. It can interface with IBM Spectrum Scale to provide cluster information/status and performance monitoring. IBM Spectrum Control uses port 22 (SSH) to interface with a single node of the IBM Spectrum Scale cluster. This node must be able to send and receive commands to all other nodes within the cluster to relay cluster state information back to the IBM Spectrum Control interface. For more information, see Table 12-2 on page 77.

IBM Spectrum Control performance data collection occurs through port 9084. For more information about this port, see Table 12-4 on page 79.

IBM Spectrum Control can also collect information about an Object/Swift configuration within IBM Spectrum Scale. Interfacing with Object/Swift requires access to ports 5000, 8080, and 35357. For more information about these ports, see Table 12-8 on page 83.

12.13 Key server ports

Although key servers are not part of IBM Spectrum Scale, and might not be inside the cluster, they should be configured such that the required ports are reachable from the nodes in the cluster.

The IBM Security Key Lifecycle Manager ports apply for both file encryption and TCT, and the DSM port number applies only to IBM Spectrum Scale encryption because TCT does not support Vormetric DSM.

Table 12-12 shows the IBM Security Key Lifecycle Manager ports.

Table 12-12 IBM Security Key Lifecycle Manager ports

Port number	Protocol	Service name	Components that are involved in communication
9083	TCP	IBM WebSphere Application Server	The `mmsklmconfig` command for retrieving a server certificate chain.
9080 (default HTTPS port to access ISKLM GUI and REST API services for SKLM 2.5 and SKLM 2.6 443 (default HTTPS port access ISKLM GUI and REST API services for SKLM 2.7 and 3.0)	TCP	IBM Security Key Lifecycle Manager REST admin interface	The `mmsklmconfig` utility for configuring IBM Spectrum Scale. The `mmkeyserv` utility which is the simplified setup tool that calls `tssklmconfig` to communicate with the key server.
5696	TCP	IBM Security Key Lifecycle Manager Key Management Interoperability Protocol (KMIP) interface	IBM Spectrum Scale daemon for retrieving encryption keys, the `mmsklmconfig` utility for configuring IBM Spectrum Scale.

Table 12-13 shows the Vormetric DSM ports.

Table 12-13 Vormetric Data Security Manager ports

Port number	Protocol	Service name	Components that are involved in communication
8445	TCP	DSM admin web GUI	The `mmsklmconfig` command for retrieving server certificate chain
5696	TCP	DSM KMIP interface	IBM Spectrum Scale daemon for retrieving encryption keys

12.14 References

The following websites are useful for further research:

- Network Configuration:

  https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/General%20Parallel%20File%20System%20%28GPFS%29/page/Network%20Configuration

- Protocols Quick Overview for IBM Spectrum Scale:

  https://ibm.biz/Bd4BTS

- Securing the IBM Spectrum Scale system by using a firewall:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_firewall.htm

- IBM Spectrum Scale port usage:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_pmpctp.htm

- Base IBM Spectrum Protect client configuration files for IBM Spectrum Scale usage:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_tsm_clientconfig.htm

- Accessing a remote GPFS file system:

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_admmcch.htm

- *Red Hat Enterprise Linux 7 Security Guide: Using Firewalls*:

  https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/sec-Using_Firewalls.html

- Ubuntu Security documentation: Firewalls:

  https://help.ubuntu.com

- Debian Wiki: Uncomplicated Firewall:

  https://wiki.debian.org/Uncomplicated%20Firewall%20%28ufw%29

- Microsoft Technet: Windows Firewall:

  https://technet.microsoft.com/en-us/network/bb545423.aspx

A

Examples of how to open firewall ports

This appendix gives examples of how to open firewall ports in various operating systems when implementing IBM Spectrum Scale.

Although the most basic examples open firewall ports on all networks, this configuration might be considered insecure for many implementations. If a higher level of security is wanted, it is necessary to restrict port traffic to only the required network or adapters. For opening ports only to specific subnets or specific adapters, see the examples in “The iptables option” on page 91.

Red Hat 7.x

You can use the following commands in Red Hat 7.x:

- List currently open ports:

```
firewall-cmd --list-ports
```

- List zones:

```
firewall-cmd --get-zones
```

- List the zone containing eth0:

```
firewall-cmd --get-zone-of-interface=eth0
```

- Opens port 1191 for TCP traffic:

```
firewall-cmd --add-port 1191/tcp
```

- Opens port 1191 for TCP traffic after a restart. Use this command to make changes persistent:

```
firewall-cmd --permanent --add-port 1191/tcp
```

- Opens a range a range of ports:

```
firewall-cmd --permanent --add-port 60000-61000/tcp
```

- Turns off/on the firewall:

```
systemctl stop firewalld
systemctl start firewalld
```

SLES12

You can use the following command in SLES12:

- Start the firewall configuration utility:

```
yast firewall
```

Ubuntu and Debian

You can use these commands in Ubuntu and Debian:

- Open port 1191 for TCP traffic:

```
sudo ufw allow 1191/tcp
```

- Open a range of ports:

```
sudo ufw allow 60000-61000/tcp
```

- Turn off/on the Uncomplicated Firewall:

```
sudo ufw disable
sudo ufw enable
```

Windows 2008R2

To find the Firewall utility In Windows, click **Control Panel** → **Administrative Tools** → **Windows Firewall with Advanced Security**. Add new rules for **Inbound / Outbound Rules** as necessary.

The iptables option

Most Linux distributions use the **`iptables`** option to set firewall rules and policies. Before using these commands, check to see which firewall zones can be enabled by default. Depending on the zone setup, the `INPUT` and `OUTPUT` terms might need to be renamed to match a zone for the wanted rule. See the Red Hat 7.x example below for one such case. You can use these commands:

- List the current firewall policies:

```
sudo iptables -S
sudo iptables -L
```

- Open port 1191 (GPFS) for inbound TCP traffic from internal subnet `172.31.1.0/24`:

```
sudo iptables -A INPUT -p tcp -s 172.31.1.0/24 --dport 1191 -j ACCEPT
```

- Open port 1191 (GPFS) for outbound TCP traffic to internal subnet `172.31.1.0/24`:

```
sudo iptables -A OUTPUT -p tcp -d 172.31.1.0/24 --sport 1191 -j ACCEPT
```

- Open port 445 (SMB) for outbound TCP traffic to external subnet `10.11.1.0/24` and only for adapter eth1:

```
sudo iptables -A OUTPUT -o eth1 -p tcp -d 10.11.1.0/24 --sport 445 -j ACCEPT
```

- Open port 445 (SMB) for inbound TCP traffic to a range of CES IPs (`10.11.1.5` - `10.11.1.11`) and only for adapter eth1:

```
sudo iptables -A INPUT -i eth1 -p tcp -m iprange --dst-range
10.11.1.5-10.11.1.11 --dport 445 -j ACCEPT
```

- Allow an internal network, eth1, to talk to an external network, eth0:

```
sudo iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
```

- Red Hat 7.x specific. Opens Chef port 8889 for inbound traffic from subnet `10.18.0.0/24` on eth1 within the public zone:

```
iptables -A IN_public_allow -i eth1 -p tcp -s 10.18.0.0/24 --dport 8889 -j
ACCEPT
```

- Save firewall rule changes to persist across a restart:

```
sudo iptables-save
```

Glossary

access control list (ACL) A list of permissions that are attached to an object. An ACL describes which users are allowed access to an object, and what operations are allowed on the object.

ACL See *access control list*.

Active Directory (AD) Directory service that Microsoft developed for Windows domain networks. It is included in most Windows Server operating systems as a set of processes and services.

AD See *Active Directory*.

Advanced Encryption Standard (AES) Specification for a symmetric-key algorithm that is established by NIST and used worldwide, superseding the Data Encryption Standard (DES). AES uses a block size of 128 bits, and a key size of 128, 192, or 256 bits.

AES See *Advanced Encryption Standard*.

Amazon Elastic Compute Cloud (EC2) A web service that provides resizable compute capacity in the cloud.

Amazon Simple Storage Service (S3) A web service that is offered by Amazon Web Services. Amazon S3 provides storage through web services interfaces.

CA See *certification authority*.

CBC See *Cipher Block Chaining*.

certification authority (CA) An entity that issues digital certificates. A digital certificate certifies the ownership of a public key by the named subject of the certificate. Trusted certificates can be used to create secure connections to a server.

Cipher Block Chaining (CBC) One of the existing "modes of operation" of cryptographic algorithms. Although a block cipher can be used for the secure cryptographic transformation of one fixed-length group of bits called a block, a mode of operation describes how to repeatedly apply a cipher's single-block operation to transform amounts of data larger than a block. In CBC mode, each block of plaintext is XORed with the previous ciphertext block before being encrypted.

EC2 See *Amazon Elastic Compute Cloud*.

Federal Information Processing Standards (FIPS) Publicly announced standards that are developed by the United States federal government for use in computer systems by non-military government agencies and government contractors.

FEK See *file encryption key*.

file encryption key (**FEK**) A key that is used to encrypt sectors of an individual file.

FIPS See *Federal Information Processing Standards*.

Hadoop Distributed File System (HDFS) A distributed file system that provides high-performance access to data across Hadoop clusters.

hash-based message authentication code (HMAC) A specific type of message authentication code (MAC) involving a cryptographic hash function in combination with a secret cryptographic key. It can be used to simultaneously verify both the data integrity and the authentication of (the sender of) a message.

HDFS See *Hadoop Distributed File System*.

HMAC See *hash-based message authentication code*.

Kerberos principal Represents a unique identity in a Kerberos system to which Kerberos can assign tickets to access Kerberos-aware services.

Key Management Interoperability Protocol (KMIP) A communication protocol that defines message formats for the manipulation of cryptographic keys on a key management server.

KMIP See *Key Management Interoperability Protocol*.

LDAP See *Lightweight Directory Access Protocol*.

Lightweight Directory Access Protocol (LDAP) A software protocol for locating organizations, individuals, and other resources, such as files and devices in a network, whether on the public internet or on a corporate intranet.

Master Encryption Key (MEK) A key that is used to encrypt a file's FEK (see *FEK*). MEKs are stored in remote key management (RKM) servers. Encryption rules that are present in the encryption policy define which MEKs should be used for a given file.

MEK See *master encryption key*.

National Institute of Standards and Technology (NIST) A measurement standards laboratory, and a non-regulatory agency of the United States Department of Commerce. NIST Special Publication 800-131A provides specific guidance for transitions to the use of stronger cryptographic keys and more robust algorithms.

Network File System (NFS) A distributed file system protocol that allows a user on a client computer to access files over a computer network much like local storage is accessed.

NFS See *Network File System*.

NIST See *National Institute of Standards and Technology*.

remote key management (RKM) RKM servers store encryption keys (see *MEK*), which are retrieved whenever files are created or opened. Supported RKM servers are IBM Security Key Lifecycle Manager and Vormetric DSM.

Remote Procedure Call (RPC) A request, possibly sent through the network, to process a subroutine in another address space. RPCs are extensively used in IBM Spectrum Scale to transmit data and requests across the network.

RKM See *remote key management*.

RPC See *Remote Procedure Call*.

S3 See *Amazon Simple Storage Service*.

Secure Sockets Layer (SSL) A standard security technology for establishing an encrypted link between a server and a client. Typically SSL (and its successor, TLS) is used to protect information that is exchanged between a web server and a browser, but it is also used to provide authentication and encryption of data that is exchanged within a cluster or across clusters.

Server Message Block (SMB) Application-layer network protocol that is mainly used for providing shared access to files, printers, and serial ports, and miscellaneous communications between nodes on a network. It also provides an authenticated inter-process communication mechanism. The most common use of SMB involves computers running Microsoft Windows.

SMB See *Server Message Block*.

SSL See *Secure Sockets Layer*.

sudo A program for UNIX like computer operating systems that allows users to run programs with the security privileges of another user, which by default is the superuser.

XEX-based tweaked-codebook mode with ciphertext stealing (**XTS**) One of the existing “modes of operation” of cryptographic algorithms, such as CBC (see *CBC*). XTS supports encrypting data whose size is not divisible by the block size.

XTS See *XEX-based tweaked-codebook mode with ciphertext stealing*.

Related publications

The publications that are listed in this section are considered suitable for a more detailed description of the topics that are covered in this paper.

IBM Redbooks

The following IBM Redbooks publication provides additional information about the topic in this document. It might be available in softcopy only.

IBM Spectrum Scale in an OpenStack Environment, REDP-5331

You can search for, view, download, or order these documents and other Redbooks, Redpapers, web docs, draft and additional materials, at the following website:

ibm.com/redbooks

Other publications

The following publications are also relevant as further information sources:

- *IBM Spectrum Scale V5.0.1: Concepts, Planning, and Installation Guide*, GC27-9262
- *IBM Spectrum Scale V5.0.1: Administration Guide*, SC27-9263
- *IBM Spectrum Scale V5.0.1: Problem Determination Guide*, SC27-9264
- *IBM Spectrum Scale V5.0.1: Command and Programming Reference*, SC27-9265
- *IBM Spectrum Scale V5.0.1: Big Data and Analytics Guide*, SC27-9266

Online resources

These websites are also relevant as further information sources:

- Accessing IBM Spectrum Scale over NFS Kerberos by using LDAP and MIT KDC

  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD106395

- Accessing a remote GPFS file system

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_admmcch.htm

- Accessing a remote IBM Spectrum Scale file system

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_admmcch.htm

- The adminMode configuration attribute

  https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_adminmodeparm.htm

- Authentication considerations

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1ins_authconcept.htm

- Authorization limitations

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_fileauthlimitations.htm

- Base IBM Spectrum Protect client configuration files for IBM Spectrum Scale usage

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_tsm_clientconfig.htm

- BigInsights 4.2.0 components

  http://www.ibm.com/support/knowledgecenter/SSPT3X_4.2.0/com.ibm.swg.im.infosphere.biginsights.install.doc/doc/c0057609.html

- BigInsights: Managing Access Control Lists (ACL) and Authorizations

  http://www.ibm.com/support/knowledgecenter/en/SSPT3X_4.2.0/com.ibm.swg.im.infosphere.biginsights.admin.doc/doc/Managing_ACLs_Authorization.html

- BigInsights: Securing the BigInsights value-added services

  http://www.ibm.com/support/knowledgecenter/en/SSPT3X_4.2.0/com.ibm.swg.im.infosphere.biginsights.admin.doc/doc/admin_val_add_sec.html

- BigInsights: Securing IBM Open Platform with Apache Spark and Apache Hadoop

  http://www.ibm.com/support/knowledgecenter/en/SSPT3X_4.2.0/com.ibm.swg.im.infosphere.biginsights.admin.doc/doc/admin_iop_sec.html

- Blog: IBM Spectrum Scale immutability function

  https://www.ibm.com/developerworks/community/blogs/storageneers/entry/Insight_to_the_IBM_Spectrum_Scale_GPFS_Immutability_function?lang=en

- Blog: Managing immutability through NFS exports and SMB shares

  https://ibm.biz/BdspSp

- Changing the object base configuration to enable S3 API

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_ChangeconfigurationenableS3.htm

- Command reference: **`mmnfs`** command

  https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_mmnfs.htm

- Configuring encryption with IBM Security Key Lifecycle Manager: Regular setup

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_encryptionenv_regular.htm

- Configuring IBM Spectrum Scale Swift and Keystone with HAProxy

  https://mkguru.wordpress.com/2015/08/12/configuring-spectrum-scale-swift-and-keystone-with-haproxy/

- Configuring OpenStack EC2 credentials

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_ConfigureOpenstackEC2credentials.htm

- Configuring sudo-wrappers in an IBM Spectrum Scale cluster

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_sudowrapper.htm

- Clipper Group: Disk and tape total cost of ownership study

  http://www.clipper.com/research/TCG2015006.pdf

- “Creating a cloud storage account” in the IBM Spectrum Scale documentation

  https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_createacloudaccount_multi.htm

- Creating containers

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_createcontainer.htm

- Creating read ACLs to authorize object users

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_createreadacl.htm

- Creating write ACLs to authorize object users

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_writeacls.htm

- Debian Wiki: Uncomplicated Firewall

  https://wiki.debian.org/Uncomplicated%20Firewall%20%28ufw%29

- Encryption chapter in the IBM Spectrum Scale documentation

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_encryption.htm

- End-to-End checksums with IBM Spectrum Scale Native RAID

  http://www.ibm.com/support/knowledgecenter/en/SSYSP8_4.5.0/com.ibm.spectrum.scale.raid.v4r21.adm.doc/bl1adv_introe2echecksum.htm

- Getting started with the IBM Spectrum Scale S3 API support by using s3curl

  https://ibm.biz/BdspMV

- IBM Spectrum Scale port usage

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_pmpctp.htm

- GPFS and Remote Shell: White paper

  https://ibm.biz/BdspS2

- Hadoop on IBM Spectrum Scale wiki

  https://ibm.biz/BdYGbL

- HDFS Transparency Security Guide

  https://ibm.biz/BdYaTh

- IBM Client Demonstration Center

  https://www.ibm.com/systems/clientcenterdemonstrations

  > **Note:** The IBM Client Demonstration Center (for IBM Business Partners, IBM employees, and anyone with an IBM ID) provides a catalog of remote demonstrations (video or live connection) that consist of self-contained material for customer demonstrations of IBM solutions. Most of the demonstrations are provided with predefined scenarios, and some also allow for the development of new scenarios. Demonstrations can also be considered as “ready to use” material for enablement or training.

- IBM Spectrum Scale Audit Logging With Varonis DatAdvantage (For 4.2.x version of IBM Spectrum Scale only)

  https://ibm.biz/BdYGbT

- IBM Spectrum Scale (IBM developerWorks)

  https://ibm.biz/BdiAwn

- IBM Spectrum Scale Immutability - Introduction and Use Cases

  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102620

- IBM Spectrum Scale in an OpenStack cloud deployment: IBM Spectrum Scale documentation

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1ins_openstackusecase.htm

- IBM Spectrum Scale Publications: Audit messages for cluster configuration changes

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1pdg_syslog.htm

- IBM Spectrum Scale Wiki

  https://ibm.biz/BdFPR2

- Introduction to Storage Security

  http://www.snia.org/sites/default/files/Storage-Security-Intro-2.0.090909.pdf

- Kerberos in IBM Open Platform with Apache Hadoop 4.1

  http://www.ibm.com/support/knowledgecenter/en/SSPT3X_4.1.0/com.ibm.swg.im.infosphere.biginsights.admin.doc/doc/admin_kerb_container.html

- Kerberos in IBM Open Platform with Apache Spark and Apache Hadoop

  http://www.ibm.com/support/knowledgecenter/en/SSPT3X_4.2.0/com.ibm.swg.im.infosphere.biginsights.admin.doc/doc/admin_kerb_container.html

- KPMG Software certificate

  https://www.kpmg.de/bescheinigungen/RequestReport.aspx?41743

- Managing GUI administrators

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_manageguiadmins.htm

- Managing OpenStack access control lists by using S3 API

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_ManagingOpenStackACLsviaAmazonS3API.htm

- Microsoft Technet: Windows Firewall

https://technet.microsoft.com/en-us/network/bb545423.aspx

- Network Configuration

  https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/General%20Parallel%20File%20System%20%28GPFS%29/page/Network%20Configuration

- OpenStack Security Guide

  http://docs.openstack.org/security-guide/

- “Preparing for encryption” in the IBM Spectrum Scale documentation

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_encryption_prep.htm

- Protocols Quick Overview for IBM Spectrum Scale

  https://ibm.biz/Bd4BTS

- Red Hat Enterprise Linux 7 Security Guide: Using Firewalls

  https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/sec-Using_Firewalls.html

- Running IBM Spectrum Scale without remote root login

  http://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_sudowrapper.htm

- SEC Rule 17a-4(f) Assessment Report

  http://www.kpmg.de/bescheinigungen/RequestReport.aspx?41742

- Securing the IBM Spectrum Scale system by using firewall

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_firewall.htm

- Securing NFS data transfer

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_configprotdatasec.htm

- Security modes

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adm_securitymode.htm

- The Storage Community

  https://developer.ibm.com/storage/blog/

- Ubuntu Security documentation: Firewalls

  https://help.ubuntu.com/

- Using multiple security levels for remote access

  http://www.ibm.com/support/knowledgecenter/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_multsecl.htm

- Varonis DatAdvantage

  https://www.varonis.com/products/datadvantage/

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

REDP-5426-01

ISBN 0738457167

Printed in U.S.A.