

IBM Storage Scale: Encryption

Phillip Gerrard

Luis Bolinches

Mika Heino

Sukumar Vankadhara



Security

Storage



IBM Redbooks

IBM Storage Scale: Encryption

March 2024

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

First Edition (March 2024)

This edition applies to IBM Storage Scale 5.1.8.0 and IBM Storage Scale System 6.1.8.3 on IBM Storage Scale System 3500.

This document was created or updated on March 15, 2024.

© Copyright International Business Machines Corporation 2024. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	v
Trademarks	vi
Preface	vii
Authors	vii
Now you can become a published author, too!	viii
Comments welcome	viii
Stay connected to IBM Redbooks	viii
Chapter 1. Introduction to Encryption in IBM Storage Scale	1
1.1 Introduction to encryption with IBM Storage Scale	2
1.2 Implementing encryption on IBM Storage Scale file systems	3
1.3 IBM Storage Scale Native RAID encryption implementation	5
1.4 SED configuration overview with IBM Storage Scale	5
1.4.1 GNR encryption summary	7
1.4.2 External Key Server interaction	7
1.5 Understanding the differences between file system and GNR encryption	8
Chapter 2. Installing IBM Security Guardium Key Lifecycle Manager	9
2.1 GKLM Server GKLM01 installation	10
2.2 GKLM Server GKLM02 installation	15
2.3 GKLM web-based GUI	21
2.3.1 GKLM Server License Activation	22
2.3.2 GKLM Server Certificate Creation	23
2.4 GKLM server replication setup	25
2.4.1 Backing up and copying the master server configuration	26
2.4.2 Restoring master server configuration to the clone server	28
2.4.3 Configuring replication on the master server	30
2.4.4 Configuring replication on the CLONE server	32
2.4.5 Confirming replication is working	33
2.5 GKLM Server Users	33
2.6 GKLM Servers and Clients	34
Chapter 3. IBM Storage Scale file system and encryption configuration	37
3.1 IBM Storage Scale clients	38
3.2 GKLM Server configuration	39
3.2.1 Configuration steps for the example environment	39
3.3 IBM Storage Scale configuration	41
3.3.1 RB_Storage_Cluster.cloud.stg.forum.ibm.com cluster	41
3.3.2 RB_Remote_Cluster.cloud.stg.forum.ibm.com cluster	43
3.3.3 RB_Muumilaakso.cloud.stg.forum.ibm.com cluster	45
3.3.4 Confirming configuration on cluster RB_Storage_Cluster	47
3.3.5 Confirming configuration on cluster RB_Remote_Cluster	48
3.3.6 Confirming configuration on IBM Storage Scale cluster RB_Muumilaakso.cloud.stg.forum.ibm.com	49
3.4 Configuring Remote clusters and remote cluster mounts	50
3.5 Enabling encryption on selected file systems and filesets	54
3.5.1 Creating encryption policy for a fileset	55
3.5.2 Checking the installed policy for a file system	56

3.5.3 Testing and changing the policy of a file system.	57
3.5.4 Verifying and testing the encryption policy	58
3.5.5 Changing a policy to include encryption for filesets	60
3.5.6 File system and fileset access summary.	62
3.6 File system and file access and attributes of encrypted files	63
Chapter 4. GNR encryption and disk hospital	65
4.1 Enabling encryption on IBM Storage Scale system	66
4.2 Disabling encryption on the IBM Storage Scale system	67
4.3 Physical disk diagnosis in an SED enabled system	68
4.4 Replacing physical disks	68
4.5 Adding physical disks	68
4.6 Deleting physical disks	69
4.7 Migration	69

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.

AIX®

Db2®

Guardium®


IBM®

IBM Security®

Power®

POWER®

Redbooks®

Redbooks (logo) ®

WebSphere®

The following terms are trademarks of other companies:

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.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

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

Red Hat is a trademark or registered trademark of Red Hat, Inc. or its subsidiaries in the United States and other countries.

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



Preface

This IBM® Redbooks® Redpaper discusses the configuration, implementation and use of encryption with IBM Storage Scale systems. It is intended to be used by technical professionals and anyone who wants to know more about encryption with IBM Storage Scale systems or anyone who is hands on with the software and devices.

Authors

Phillip Gerrard is a Project Leader for the International Technical Support Organization working out of Beaverton, Oregon. As part of IBM for over 15 years he has authored and contributed to hundreds of technical documents published to IBM.com and worked directly with IBM's largest customers to resolve critical situations. As a team lead and Subject Matter Expert for the IBM Storage Protect support team, he is experienced in leading and growing international teams of talented IBMers, developing and implementing team processes as well as creating and delivering education. Phillip holds a degree in computer science and business administration from Oregon State University.

Luis Bolinches is part of the IBM Storage Scale development team. He has been working with Scale since version 3.4 and with ESS prior to GA. With a background of Networking, Power® systems and Linux, he is regularly involved with large customer engagements, development of ESS deployment solutions, and customer facing events.

Mika Heino is a senior IT Management Consultant with IBM Lab Services working in Finland for local and international IBM accounts. He has a degree in Telecommunications and Computer Science from Turku University of Applied Sciences. Mika has 25 years experience with Linux, IBM AIX® and IBM i, and server virtualization for both Intel and IBM POWER®. He is a Master Certified Technical Specialist for Storage Systems with more than 15 years of experience with storage area networks (SANs), IBM Storage Systems, and storage virtualization.

Sukumar Vankadhara is a Staff Software Engineer working in India and specializes in IBM Storage Scale Native RAID, IBM Storage Scale System and Erasure Code Edition testing.

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 IBM Redbooks® 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 found at:

ibm.com/redbooks

- Send your comments in an email to:

redbooks@us.ibm.com

- Mail your comments to:

IBM Corporation, IBM Redbooks
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on LinkedIn:

<https://www.linkedin.com/groups/2130806>

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

<https://www.redbooks.ibm.com/subscribe>

- Stay current on recent Redbooks publications with RSS Feeds:

<https://www.redbooks.ibm.com/rss.html>



Introduction to Encryption in IBM Storage Scale

This chapter provides an overview of encryption within IBM Storage Scale. The target audience is familiar with IBM Storage Scale basics, can create file systems and filesets, and has a basic understanding of the policy engine to understand encryption in an IBM Storage Scale environment.

The following topics are presented in this chapter:

- ▶ 1.1, “Introduction to encryption with IBM Storage Scale” on page 2
- ▶ 1.2, “Implementing encryption on IBM Storage Scale file systems” on page 3
- ▶ 1.3, “IBM Storage Scale Native RAID encryption implementation” on page 5
- ▶ 1.4, “SED configuration overview with IBM Storage Scale” on page 5
- ▶ 1.5, “Understanding the differences between file system and GNR encryption” on page 8

1.1 Introduction to encryption with IBM Storage Scale

There are two methods for implementing encryption at rest in an IBM Storage Scale environment:

1. At the file system level, encryption and decryption happen directly on the clients of a Storage Scale system.
2. At the IBM Storage Scale Server, encryption and decryption happen on the storage back-end server.

This publication explains both of these options and provides examples.

Storage Scale clusters can span global environments and consist of various disk technologies. There is a need to protect this data against unauthorized access regardless of where it is stored. Beginning with General Parallel File System (GPFS) 4.1, the encryption feature is available for configuration and use. Encryption is implemented in Storage Scale file systems with the following features:

- ▶ No special requirements for various storage hardware
- ▶ Generates finer levels of security granularity by cascading different security levels by using specific keys
- ▶ Provides Federal Information Processing Standard (FIPS)-compliant encryption

Data *at rest* means that the data is encrypted on disk and is decrypted by the Storage Scale layer before it is sent to the requester. Protection of data at rest uses encryption to make data unreadable to any party who does not possess the necessary decryption keys.

Encryption is often paired together with secure data deletion for increased data security. When you use modern storage hardware, it is difficult, and sometimes impossible for SSDs, to verify that a piece of data is deleted, overwritten, or made unavailable or inaccessible to a cyberattacker. Secure data deletion allows an extra layer of security by removing encryption data from the remote key server, which effectively makes the data unreadable even if it is accessed.

The use of encryption at the file system level does not actively protect against network traffic snooping. If data confidentiality and possible traffic snooping is a concern, the use of encryption at the scale server level might be a better choice. One beneficial side effect of encrypting data blocks before the node sends them to a logical block device or an NSD server is that it prevents an attacker from reading the unencrypted network traffic.

The proposed solution does not claim to defend against an active network cyberattacker that modifies or crafts its own packets. The cyberattacker might, for example, compromise another Storage Scale node or delete data that belongs to other tenants. In these cases, the use of a cipherList can be used to help defend against an active network attacker.

The most common use case that is addressed by the encryption feature is that some customers require a Federal Information Processing Standard (FIPS) 140-2 compliant solution. The implementation procedure that is provided in this IBM Redbooks publication describes the steps to build a FIPS-compliant solution. The GSKit that is included with Storage Scale is FIPS certified, and it is used by Storage Scale for file system crypto-related functions.

This chapter helps to provide details about the configuration procedure for an encryption capable Storage Scale environment and provides common-use encryption cases. For more

details and other theoretical and background information, see [IBM Storage Scale: Advanced Administration Guide](#).

Note: Encryption is only supported on IBM AIX, Linux x86_64, and Linux ppc64le. Encryption on a Windows operating system is not supported as of the writing of this IBM Redbooks publication.

1.2 Implementing encryption on IBM Storage Scale file systems

Encryption processing with IBM Storage Scale consists of three major components:

1. A Storage Scale cluster that is running a supported version of Storage Scale
2. A remote key manager (RKM) for providing access control to keys

Note: For more information and for updates about supported RKM versions, see [IBM Storage Scale Overview](#).

3. Encryption rules by Storage Scale policies

Every file is encrypted and decrypted using its file encryption key (FEK). The i-node metadata for the file system is not encrypted.

The FEK is stored and encrypted in the extended attribute (EA) in the i-node.

The FEK is used for encrypting and decrypting the data on disk and is not accessible directly.

All FEKs are encrypted with a master encryption key (MEK) when stored in the i-node.

An FEK cannot be stored unencrypted and is encrypted with at least one MEK. Therefore, to access the FEK, you must have access to all the MEKs that were used to encrypt the FEK.

It is possible to wrap an FEK up to eight times with any other specific combination of MEKs. See Figure 1-1 on page 4.

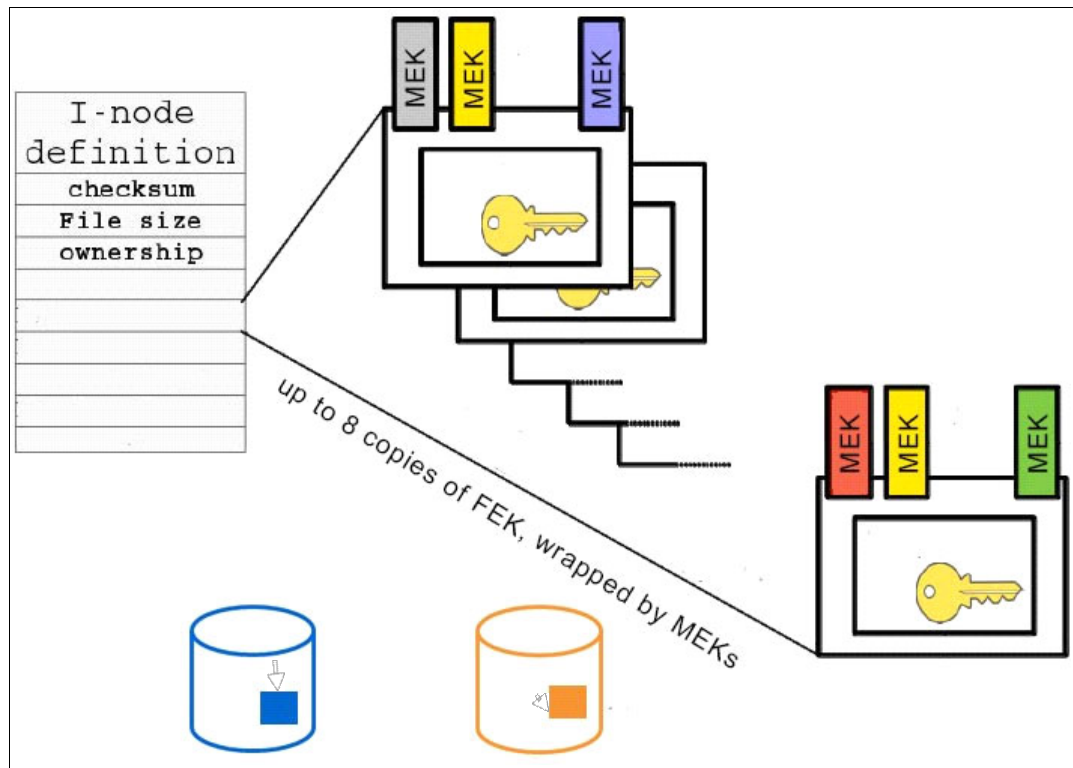


Figure 1-1 Mode information with MEK/FEK

Note: It is possible to wrap an FEK up to eight times with any other specific combination of MEKs.

The Remote Key Manager (RKM) must be from a supported key server within the Storage Scale environment. In this publication, an IBM Security® Key Lifecycle Manager (SKLM) is configured and shows how Storage Scale uses it as an RKM. This includes showing how a Storage Scale node gets the MEK from the RKM by authorizing itself with a certificate called *keystore*.

Separating the key management from the Storage Scale cluster allows for a higher level of security by managing the keys on a system outside of Storage Scale. By separating the administrator rights, you can ensure that even an administrator with superuser rights from inside the cluster cannot change the key configuration on the remote key management system.

This approach is referred to as the “Four eyes principle” or “Two persons rule”.

Whichever supported RKM is chosen, always use a highly available (HA) configuration. If the access to RKM is lost, it likely has a negative impact on data accessibility. It is also tremendously important to back up the MEK. Losing the keys on the MEK can cause access issues and cause data to no longer be viable. This IBM Redbooks publication provides an example that shows how to set up an HA SKLM as the RKM.

Because MEKs are stored remotely, access to the MEKs is managed on a per node basis with certificates. These certificates are held and accessed out of a keystore of a node. This means that a node is required to have a valid certificate to get access to the MEK.

Note: When an MEK is read and is known to the Storage Scale node, it remains in the Storage Scale daemon's memory. Therefore, there is no performance penalty for storing the MEK remotely.

The policy rules that are defined in the Storage Scale environment determine whether a file gets encrypted or not. The encryption state of a file is determined at file creation time by encryption-specific Storage Scale file placement policy rules. A maximum of eight rules can be applied per file. All encryption and decryption operations happen in the node that is performing the write and read operations of the file.

The primary goal for using encryption in an IBM Storage Scale environment is to ensure that the data at rest is protected from unauthorized access. An individual might gain physical access to a disk removed from the environment or gain the ability to access data from nodes that are not explicitly allowed. Because the at rest data is encrypted, the data is not in a readable format, which renders the data useless to that individual.

IBM Storage Scale encryption can also be viewed as a way to control access to the data in a file system with the use of keys. This combined with multi-clusters provides a method of access control for data in a shared environment. This example is also covered within this IBM Redbooks publication.

1.3 IBM Storage Scale Native RAID encryption implementation

An addition to encryption in IBM Storage Scale is the ability to use self-encrypting drives (SEDs) in GPFS Native RAID (GNR). Both file system and SED encryption can be used separately or in parallel. This publication provides information on both configurations and a comparison so that you can make an informed decision on which method or methods best fit your needs.

The SED provides hardware-based encryption, which protects the data at rest by locking the drives while the power is off. Also, SED also supports hardware-based crypto erase. GNR uses these SED capabilities to build and manage drives and provides encryption at rest capabilities for all GNR data.

Note: Read the product documentation to confirm which IBM Storage Scale Server configurations are supported for encryption within GNR.

The use of SED features with GNR includes the following benefits:

- ▶ Use of Cryptographic Erase for disposal of disks
- ▶ Instantaneous change of Authentication key (rekey) without loss of data
- ▶ Encryption of all data with minimal performance degradation
- ▶ Simplified key management
- ▶ Only signed firmware can be loaded onto an SED

1.4 SED configuration overview with IBM Storage Scale

IBM Storage Scale supports SED features and provides data-at-rest encryption by using an MEK. The Master Encryption Key is stored in the RKM. The MEK is used by GNR for locking

all the drives in use within the GPFS cluster. In this publication, the same SKLM server is configured and serves as the RKM for GNR encryption.

Note: Always check product documentation for details on which key servers are supported for GNR encryption.

This MEK is sensitive and requires careful handling. A compromised MEK can potentially compromise all data that is protected by this key, and the loss of an MEK is equivalent to losing access to all drives encrypted using this MEK. It is an IBM policy that if IBM facilities are used to store encrypted data belonging to a customer, the responsibility of encryption key management lies with the customer. IBM does not store the encryption key.

When an SED is removed from its owner's system it loses power, and its recorded data is locked against unauthorized access. When power is restored, the host system must prove that it owns the drive. This is done by providing the drive with the appropriate ownership credentials or password. These credentials are provided to the drive through the MEK for the GPFS cluster.

The following Figure 1-2 shows an overview of the components that are involved in GNR encryption.

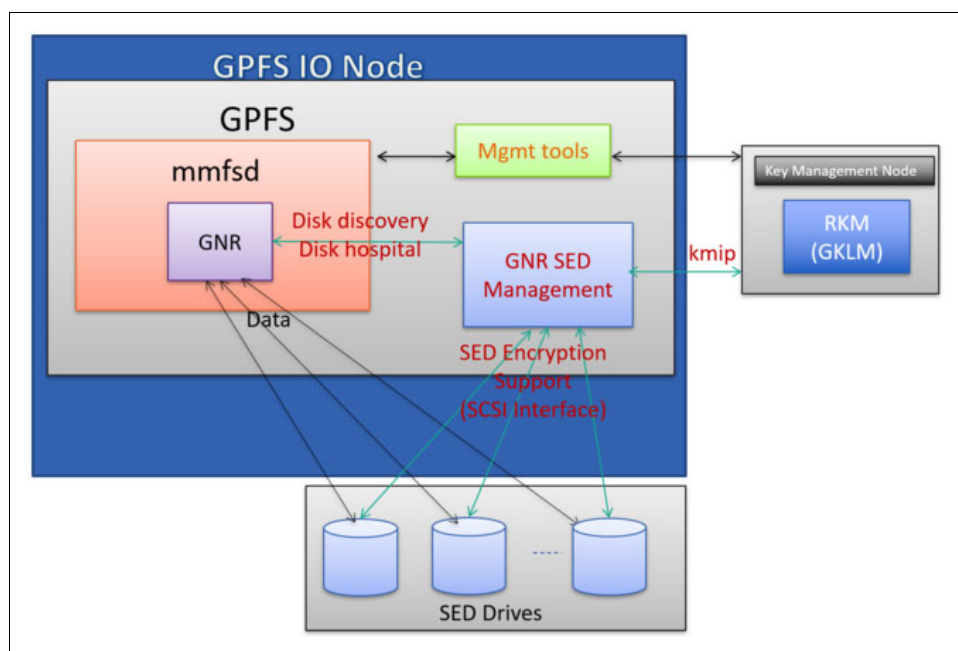


Figure 1-2 GNR encryption overview diagram

Figure 1-2 shows the main components of the system:

- ▶ GNR
 - GPFS Native RAID component
- ▶ Management tools
 - A set of command-line tools to set up and manage operations that are not triggered by standard FS I/O operations such as mmchcarrier and mmaddpdisk

- ▶ Remote Key Manager (RKM)
Responsible for managing and storing durable copies of keys and can be hosted by the tenant or by IBM
- ▶ GNR SED Management (mmvdisk sed)
Utility responsible for getting the keys from the external key server, locking and unlocking of SEDs, and handling callouts from the GNR Disk Hospital

1.4.1 GNR encryption summary

The following list provides the key features and concepts of GNR encryption:

- ▶ All user data is written to self-encrypting HDDs and SSDs in the IBM Storage Scale System storage enclosures.
- ▶ The data that GNR writes to its log-tip is not encrypted unless log-tip is configured from self-encrypting HDD or NVMe drives.
- ▶ The ability to present the Key to the disks requires access to an external unique Key Server. This means that access to this key must be provided upon system start-up from outside GNR.
- ▶ Rekeying is supported. This is defined as changing one or more keys.
- ▶ The user is able to switch encryption from on to off. Each transition from on to off makes all existing data unreadable. Switching encryption off securely erases all data on the GNR system, as the encryption keys on all disks are replaced. This is also referred to as crypto erase.
- ▶ Adding and replacing disks and modules can be done with encryption enabled or disabled. The new or replaced unit is configured in the same state as the rest of the system. The addition and replacement of disks that do not support encryption fails.
- ▶ Crypto erase is the ability to effectively erase all user data by changing the actual encryption keys on the disks.
- ▶ If SED support is enabled for IBM Storage Scale System, then all the drives, including the SSDs in the recovery groups should be SED capable drives.

1.4.2 External Key Server interaction

The `mmkeyserv` command supports the following actions for a user of GPFS:

- ▶ Verify the GKLM server configuration
- ▶ Create a device group on the GKLM server for owning and managing the keys
- ▶ Create clients with credentials for communicating with the GKLM server and retrieving keys
- ▶ Create one or more cryptographic keys on the GKLM server
- ▶ Retrieve the unique identifiers of the created keys
- ▶ Remove expired or compromised keys

1.5 Understanding the differences between file system and GNR encryption

File system encryption and GNR encryption can both achieve the same goal of providing encryption to data while at rest on a storage device. However, they each have important differences and requirements that need to be understood before you choose which methods to implement:

- ▶ You are required to have an IBM Storage Scale server, formally known as ESS, with supported hardware and a software level that is able to use GNR encryption. It can be mixed with non-Scale storage systems, but only the Storage Scale server supports encryption at rest. File system encryption can work with any type of storage, but it requires extra CPU utilization on each of the clients.
- ▶ GNR encryption can be configured and used with any IBM Storage Scale edition. Configuring encryption at the file system level requires the use of IBM Storage Scale Data Management Edition (DME).
- ▶ CPU usage for GNR encryption is minimal and typically has close to a nonmeasurable impact on performance. File system encryption is done by the CPU of the client node and is distributed across clients, so although an individual client might show an impact, the back end overall would not.
- ▶ GNR encryption is a box-level feature, regardless of what file systems are used. File system encryption provides per node, fileset, and file system granularity.
- ▶ GNR encryption is encrypted and decrypted at the drive level and travels from the backend server, over the network decrypted. When using file system encryption, file data network traffic is encrypted before it is transmitted through the network.
- ▶ No file system configuration is needed to implement GNR encryption. With file system-based encryption, additional configuration steps are required and is covered later in this publication.
- ▶ GNR encryption encrypts all data at rest, including attributes. File system level encryption encrypts the file data but not the metadata for the file, which includes the file name, access time and other attributes. Sometimes, both encryption methods can be used to achieve certain compliance requirements.

When selecting which encryption method to use for securing data, these examples are just a few scenarios to consider. This is not a comprehensive list of all the possible points to be considered before making a final decision.



Installing IBM Security Guardium Key Lifecycle Manager

This chapter provides detailed instructions to install and configure IBM Security Guardium® Key Lifecycle Manager (GKLM) redundant servers in a master-clone configuration using Red Hat Enterprise Linux RHEL 8.8. This includes the configuration steps for IBM Storage Scale clients to enable encryption.

This chapter includes a detailed step-by-step procedure to set up an environment for running an encrypted file system.

The following topics are presented in this chapter:

- ▶ 2.1, “GKLM Server GKLM01 installation” on page 10
- ▶ 2.2, “GKLM Server GKLM02 installation” on page 15
- ▶ 2.3, “GKLM web-based GUI” on page 21
- ▶ 2.4, “GKLM server replication setup” on page 25
- ▶ 2.5, “GKLM Server Users” on page 33
- ▶ 2.6, “GKLM Servers and Clients” on page 34

2.1 GKLM Server GKLM01 installation

The following section contains examples that demonstrate a step by step configuration of IBM GKLM Server, GKLM01, on RHEL 8.8 from installation media. The Red Hat installation type used for OS installation was a minimal installation without any additional packages selected.

For a successful installation of the GKLM server on an RHEL 8.8 host with a minimal installation, some additional software packages are required. The additional packages that are installed are listed in Example 2-2 and can be found with the RHEL 8.8 OS installation media.

IBM GKLM software installation also requires some of the OS settings to be set to specified values. The settings are listed in Example 2-4.

1. Installation packages that are used for GKLM installation are stored on a local server at /root/GKLM420/. See Example 2-1.

Example 2-1

```
SGKLM_4.2.0_F_LINUX_SERVER_10F2.tar
SGKLM_4.2.0_F_LINUX_SERVER_20F2.tar
4.2.0-ISS-GKLM-FP0001-Linux.tar.gz
SGKLM_4.2.0_LICENSE_MP.zip
```

2. Additional RHEL8 packages must be installed with the minimal installation. These are required by IBM GKLM for installation. See Example 2-2.

Example 2-2

```
root@gklm01 GKLM420]# yum install ksh
[root@gklm01 GKLM420]# yum install libstdc++-8.5.0-18.el8.i686
[root@gklm01 GKLM420]# yum install pam-1.3.1-25.el8.i686
[root@gklm01 GKLM420]# yum install unzip
[root@gklm01 GKLM420]# yum install binutils
[root@gklm01 GKLM420]# yum install net-tools
```

3. Before installing GLKLM, rename the license file and decompress the installation files. See Example 2-3.

Example 2-3 Resource preparation step

```
[root@gklm01 GKLM420]# mv SGKLM_4.2.0_LICENSE_MP.zip gklm.license.zip
[root@gklm01 GKLM420]# tar -xvf SGKLM_4.2.0_F_LINUX_SERVER_10F2.tar
[root@gklm01 GKLM420]# tar -xvf SGKLM_4.2.0_F_LINUX_SERVER_20F2.tar
[root@gklm01 GKLM420]# tar -xvf 4.2.0-ISS-GKLM-FP0001-Linux.tar.gz
```

4. Change the U-mask and confirm the configured shell. Set the umask to **0022**. The bash shell must be installed. See Example 2-4.

Example 2-4

```
[root@gklm01 disk1]# umask
0022
[root@gklm01 disk1]# which bash
/usr/bin/bash
```

5. Add firewall rules to allow connection to the GKLM web-based GUI. The example uses the default ports that are used by GKLM. See Example 2-5.

Example 2-5

```
[root@gklm01 disk1]# firewall-cmd --zone=public --permanent --add-port=9443/tcp
Success
[root@gklm01 data]# firewall-cmd --zone=public --permanent --add-port=1441/tcp
success
[root@gklm01 data]# firewall-cmd --zone=public --permanent --add-port=5696/tcp
Success
[root@gklm01 data]# firewall-cmd --zone=public --permanent --add-port=1111/tcp
success
[root@gklm01 disk1]# firewall-cmd --reload
success
[root@gklm01 disk1]# firewall-cmd --list-ports
1441/tcp 2222/tcp 5696/tcp 9443/tcp
[root@gklm01 disk1]#
```

6. Verify that /tmp permissions are set to 777.

Example 2-6

```
[root@gklm01 disk1]# ls -la /tmp/
total 602144
drwxrwxrwt. 19 root    root          4096 Aug  4 17:00 .
```

7. Disable SELINUX. A reboot is required for changes to take effect. See Example 2-7.

Example 2-7

```
[root@gklm01 disk1]# vi /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are
protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

8. Create an encryption string that is used to encrypt the passwords for the internal IBM Db2® Admin and sklmAdmin users then define passwords for their use. See Example 2-8.

Example 2-8

```
[root@gklm01 /] # ./root/GKLM420/disk1/im/tools/imcl encryptString OME1por2@per3
1y232N6S6QG3PNdv1NvFbg==
[root@gklm01 /] #
```

Note: The sklmAdmin user ID is *not* case-sensitive, so other variations such as SKLMAdmin can also be used.

9. Modify the silent installation file used for GKLM server installation. See Example 2-9.

Example 2-9

```
[root@gklm01 disk1] # vi SKLM_Silent_Linux_Resp.xml
<server>
    <repository location='/root/GKLM420/disk1/im'/>
    <repository location='/root/GKLM420/disk1/'/>
</server>
<!--The DB2 Group name should not be more than 8 characters -->
<data key='user.DB2_ADMIN_PWD,com.ibm.gklm42.db2.lin.ofng
value='1y232N6S6QG3PNdv1NvFbg=='/>
<data key='user.CONFIRM_PASSWORD,com.ibm.gklm42.db2.lin.ofng'
value='1y232N6S6QG3PNdv1NvFbg=='/>
</profile>
<data key='user.SKLM_ADMIN_PASSWORD,com.ibm.gklm42.linux'
value='1y232N6S6QG3PNdv1NvFbg=='/>
<data key='user.SKLM_ADMIN_CONF_PWD,com.ibm.gklm42.linux'
value='1y232N6S6QG3PNdv1NvFbg=='/>
```

10. Perform GKLM server silent installation on host GKLM01. See Example 2-10.

Example 2-10

```
[root@gklm01 disk1]# ./silent_install.sh SKLM_Silent_Linux_Resp.xml -acceptLicense
No preinstalled IBM Installation Manager found on the system.
Installing IBM Security Guardium Key Lifecycle Manager v4.2.0.0
Fri Aug 4 12:22:51 EEST 2023 - SKLM Prerequisite check started.
Db2 pre-requisite check - PASSED.
Checking required shell - PASSED
Checking executable permissions - PASSED
Checking required umask - PASSED
Checking kernel parameters - WARNING
Checking SELinux - PASSED
Checking CPU speed - PASSED
Checking RAM - PASSED
Fri Aug 4 12:22:51 EEST 2023 - SKLM Prerequisite check - PASSED with WARNING.
The Prerequisite check passed with one or more warnings. Review the warning
details in /tmp/SKLMPreqCheck.log. Press any key to continue. Installation of IBM
Security Guardium Key Lifecycle Manager is not supported on the current operating
system. Continuing the installation on an unsupported operating system.
Installed com.ibm.cic.agent_1.9.2003.20220917_1018 to the
/opt/IBM/InstallationManager/eclipse directory.
Installed com.ibm.gklm42.db2.lin.ofng_11.5.8.0 to the /opt/IBM/DB2GKLMV42
directory.
Installed com.ibm.websphere.liberty.BASE_22.0.12.20221107_1900 to the
/opt/IBM/WebSphere/Liberty directory.
Installed com.ibm.java.jdk.v8_8.0.7020.20221026_1851 to the
/opt/IBM/WebSphere/Liberty directory.
Installed com.ibm.gklm42.linux_4.2.0.0 to the /opt/IBM/GKLMV42 directory.
CRIMA1137W WARNING: The following packages do not support the 64-bit version of
Installation Manager that you are using: IBM Db2 version 11.5.8.0, IBM Security
```

Guardium Key Lifecycle Manager version 4.2.0.0. If you continue, you might have issues with installation and deployment. For information about 64-bit mode support for a package, see the package documentation.

Explanation: The 64-bit version of Installation Manager checks each package for 64-bit support. If a package does not support the 64-bit version, you receive a warning.

User Action: Use a 32-bit version of Installation Manager to install the package.
WARNING: Support for using Java SE 7 and Java SE 7.1 with WebSphere Liberty ended. The Liberty kernel can no longer run with Java SE 7 or Java SE 7.1.
The installation process is complete. Please look into Installation Manager logs for details.

```
[root@gklm01 disk1]#
```

Note: Check the kernel parameter *kernel_msgmni* value with the command **sysctl -a | grep kernel.msgmni**. If the value is set to current RAM * 1024 (in GB), ignore the warning on kernel parameters pre-check.

Ignore the error message not supported operating system displayed during the installation. This is an error that is returned from the install check utility for SGKLM 4.2. RHEL8.8 is listed as supported. For more information, see [IBM Security Guardium Key Lifecycle Manager Support Matrix](#).

11. Modify Db2 permissions to allow GKLM access to the database on host GKLM01. See Example 2-11.

Example 2-11

```
[root@gklm01 /]# su - klmdb42
```

```
[klmdb42@gklm01 ~]$ db2 connect to klmdb42
```

Database Connection Information

```
Database server          = DB2/LINUX8664 11.5.8.0
SQL authorization ID     = KLMDB42
Local database alias     = KLMDB42
```

```
[klmdb42@gklm01 ~]$ db2 grant secadm on database to user klmdb42
DB20000I The SQL command completed successfully.
[klmdb42@gklm01 ~]$
```

12. Backup the current configuration and install the latest FixPack for GKLM on host GKLM01. As of this writing the latest fix pack is version 4.2.0.1. See Example 2-12.

Example 2-12

```
[root@gklm01 /]# exec bash
[root@gklm01 /]# cd /opt/IBM/WebSphere/Liberty/bin/
```

```
[root@gklm01 bin]# ./stopServer.sh
```

Stopping server gklm42server.
Server gklm42server stopped.

```

[root@gklm01 bin]#

[root@gklm01 bin]# mkdir /tmp/wasbackup
[root@gklm01 bin]# cd /tmp/wasbackup/

[root@gklm01 wasbackup]# tar -cvf wasbackup.tar /opt/IBM/WebSphere/Liberty/*
tar: Removing leading `/' from member names
/opt/IBM/WebSphere/Liberty/bin/
...
<Trunkated output>
...
[root@gklm01 wasbackup]#

[root@gklm01 wasbackup]# cd /opt/IBM/WebSphere/Liberty/bin/
[root@gklm01 bin]# ./startServer.sh

Starting server gklm42server.
Server gklm42server started with process ID 57205.
[root@gklm01 bin]#

[root@gklm01 GKLM420]# cd /root/GKLM420/sklm/

[root@gklm01 sklm]# cp SKLM_Silent_Update_Linux_Resp.xml
SKLM_Silent_Update_Linux_Resp.xml.original
[root@gklm01 sklm]# vi SKLM_Silent_Update_Linux_Resp.xml

<?xml version="1.0" encoding="UTF-8"?>
<!--The "acceptLicense" attribute has been deprecated. Use "-acceptLicense"
command line option to accept license agreements.-->
<agent-input clean='true'>
<server>
<repository location='/root/GKLM420/javafp/repository.config'/>
<repository location='/root/GKLM420/wasfp/repository.config'/>
<repository location='/root/GKLM420/sklm/repository.config'/>
</server>
<profile id=' IBM Security Guardium Key Lifecycle Manager v4.2.0' installLocation='/opt/IBM/GKLMV42'>
<data key='eclipseLocation' value='/opt/IBM/GKLMV42'/>
<data key='user.SKLM_ADMIN_USER,com.ibm.gklm42.linux' value='SKLMAdmin'/>
<data key='user.SKLM_ADMIN_PASSWORD,com.ibm.gklm42.linux'
value='1y232N6S6QG3PNdv1NvFbg=='/>
<data key='user.DB_ADMIN_USER,com.ibm.gklm42.linux' value='k1mdb42'/>
<data key='user.DB_ADMIN_PASSWORD,com.ibm.gklm42.linux'
value='1y232N6S6QG3PNdv1NvFbg=='/>
</profile>
<install modify='false'>
<offering id='com.ibm.java.jdk.v8' profile='WebSphere Liberty V_220012'
installFixes='none' version='8.0.8000.20230314_0926'/>
<offering id='com.ibm.websphere.liberty.BASE' profile='WebSphere Liberty V_220012'
installFixes='none' version='23.0.3.20230319_1900'/>
<offering id='com.ibm.gklm42.linux' version='4.2.0000.1' profile='IBM Security
Guardium Key Lifecycle Manager v4.2.0' features='main.feature'
installFixes='none'/>
</install>
</agent-input>

```

13. Save the changes to SKLM_Silent_Update_Linux_Resp.xml.

Example 2-13

```
[root@gk1m02 GKLM420]# cd /root/GKLM420/
[root@gk1m01 GKLM420]# ./silent_updateSKLM.sh /opt/IBM/InstallationManager/
/opt/IBM/WebSphere/Liberty/
About to install FixPack...
Name: WebSphere Liberty V_220012
Launching InstallManager...
/opt/IBM/InstallationManager//eclipse/tools/imcl -input
/root/GKLM420/sklm/SKLM_Silent_Update_Linux_Resp.xml -silent -acceptLicense
Updated to com.ibm.java.jdk.v8_8.0.8000.20230314_0926 in the
/opt/IBM/WebSphere/Liberty directory.
Updated to com.ibm.websphere.liberty.BASE_23.0.3.20230319_1900 in the
/opt/IBM/WebSphere/Liberty directory.
Updated to com.ibm.gk1m42.linux_4.2.0.1 in the /opt/IBM/GKLMV42 directory.
CRIMA1137W WARNING: The following packages do not support the 64-bit version of
Installation Manager that you are using: IBM Security Guardium Key Lifecycle
Manager version 4.2.0.1, IBM Security Guardium Key Lifecycle Manager installed
version 4.2.0.0. If you continue, you might have issues with installation and
deployment. For information about 64-bit mode support for a package, see the
package documentation.

Explanation: The 64-bit version of Installation Manager checks each package for
64-bit support. If a package does not support the 64-bit version, you receive a
warning.

User Action: Use a 32-bit version of Installation Manager to install the package.
WARNING: Support for using Java SE 7 and Java SE 7.1 with WebSphere Liberty ended.
The Liberty kernel can no longer run with Java SE 7 or Java SE 7.1.
Installation of fix pack is complete. See the Installation Manager log files for
more information.
[root@gk1m01 GKLM420]#
```

This completes the installation of the GKLM server.

2.2 GKLM Server GKLM02 installation

The following section contains examples which demonstrate a step-by-step installation and configuration of an IBM GKLM Server, GKLM02, which will be included as part of a master-clone relationship.

The installation is performed using the operating system RHEL 8.8 from installation media. The Red Hat installation type is a minimal installation without any additional packages selected.

Additional software packages are required for successful installation of GKLM server on RHEL 8.8 minimal installation. Additional packages installed are listed in Example 2-15 on page 16 and are on RHEL 8.8 OS installation media.

IBM GKLM software installation requires some of the OS settings to be set to specified values. These settings are verified in Example 2-17 on page 16.

1. Installation packages used for GKLM installation are stored on a local server at /root/GKLM420/. See Example 2-14.

Example 2-14 Installation packages used for GKLM installation

```
SGKLM_4.2.0_F_LINUX_SERVER_10F2.tar
SGKLM_4.2.0_F_LINUX_SERVER_20F2.tar
4.2.0-ISS-GKLM-FP0001-Linux.tar.gz
SGKLM_4.2.0_LICENSE_MP.zip
```

2. Install additional packages for RHEL8 minimal installation that are required by IBM GKLM installation. See Example 2-15.

Example 2-15

```
[root@gklm02 GKLM420]# yum install ksh
[root@gklm02 GKLM420]# yum install libstdc++-8.5.0-18.el8.i686
[root@gklm02 GKLM420]# yum install pam-1.3.1-25.el8.i686
[root@gklm02 GKLM420]# yum install unzip
[root@gklm02 GKLM420]# yum install binutils
[root@gklm02 GKLM420]# yum install net-tools
```

3. Before installing GLKLM, rename the license file and decompress the installation files. See Example 2-16.

Example 2-16 GKLM install resource preparation

```
root@gklm02 GKLM420]# mv SGKLM_4.2.0_LICENSE_MP.zip gklm.license.zip
[root@gklm02 GKLM420]# tar -xvf SGKLM_4.2.0_F_LINUX_SERVER_10F2.tar
[root@gklm02 GKLM420]# tar -xvf SGKLM_4.2.0_F_LINUX_SERVER_20F2.tar
[root@gklm02 GKLM420]# tar -xvf 4.2.0-ISS-GKLM-FP0001-Linux.tar.gz
```

4. Change U-mask and confirm the configured shell. The umask must be set to 0022 and a bash shell must be installed. See Example 2-17.

Example 2-17

```
[root@gklm02 disk1]# umask
0022
[root@gklm02 disk1]# which bash
/usr/bin/bash
```

5. Add firewall rules to allow connection to the GKLM web-based GUI. The example uses the default ports that are used by GKLM. See Example 2-18.

Example 2-18

```
[root@gklm02 disk1]# firewall-cmd --zone=public --permanent --add-port=9443/tcp
Success
[root@gklm02 data]# firewall-cmd --zone=public --permanent --add-port=1441/tcp
success
[root@gklm02 data]# firewall-cmd --zone=public --permanent --add-port=5696/tcp
Success
[root@gklm02 data]# firewall-cmd --zone=public --permanent --add-port=2222/tcp
success
[root@gklm02 disk1]# firewall-cmd --reload
success
[root@gklm02 disk1]# firewall-cmd --list-ports
1441/tcp 2222/tcp 5696/tcp 9443/tcp
```

```
[root@gklm02 disk1]#
```

6. Verify that /tmp permissions are set to 777. See Example 2-19.

Example 2-19

```
[root@gklm02 disk1]# ls -la /tmp/
total 602144
drwxrwxrwt. 19 root    root          4096 Aug  4 17:00 .
```

7. Disable SELINUX. A reboot is required for changes to take effect. See Example 2-20.

Example 2-20

```
[root@gklm02 disk1]# vi /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are
protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

8. Create an encryption string that is used to encrypt the passwords for the internal Db2 Admin and sklmAdmin users then define passwords for their use. See Example 2-21.

Example 2-21

```
[root@gklm02 /] # ./root/GKLM420/disk1/im/tools/imcl encryptString OME1por2@per3
1y232N6S6QG3PNdv1NvFbg==
[root@gklm02 /] #
```

9. Modify the silent installation file used for GKLM server installation. See Example 2-22.

Example 2-22

```
[root@gklm02 disk1] # vi SKLM_Silent_Linux_Resp.xml
<server>
    <repository location='/root/GKLM420/disk1/im'/>
    <repository location='/root/GKLM420/disk1/'/>
</server>
<!--The DB2 Group name should not be more than 8 characters -->
<data key='user.DB2_ADMIN_PWD,com.ibm.gklm42.db2.lin.ofng
value='1y232N6S6QG3PNdv1NvFbg=='/>
<data key='user.CONFIRM_PASSWORD,com.ibm.gklm42.db2.lin.ofng'
value='1y232N6S6QG3PNdv1NvFbg=='/>
</profile>
<data key='user.SKLM_ADMIN_PASSWORD,com.ibm.gklm42.linux'
value='1y232N6S6QG3PNdv1NvFbg=='/>
<data key='user.SKLM_ADMIN_CONF_PWD,com.ibm.gklm42.linux'
value='1y232N6S6QG3PNdv1NvFbg=='/>
```

10. Perform a GKLCM server silent installation on host GKLM02. See Example 2-23.

Example 2-23 GKLM server silent installation

```
[root@gklm02 disk1]# ./silent_install.sh SKLM_Silent_Linux_Resp.xml -acceptLicense
No preinstalled IBM Installation Manager found on the system.
Installing IBM Security Guardium Key Lifecycle Manager v4.2.0.0
Fri Aug 4 12:22:51 EEST 2023 - SKLM Prerequisite check started.
Db2 pre-requisite check - PASSED.
Checking required shell - PASSED
Checking executable permissions - PASSED
Checking required umask - PASSED
Checking kernel parameters - WARNING
Checking SELinux - PASSED
Checking CPU speed - PASSED
Checking RAM - PASSED
Fri Aug 4 12:22:51 EEST 2023 - SKLM Prerequisite check - PASSED with WARNING.
The Prerequisite check passed with one or more warnings. Review the warning
details in /tmp/SKLMPreqCheck.log. Press any key to continue. Installation of IBM
Security Guardium Key Lifecycle Manager is not supported on the current operating
system. Continuing the installation on an unsupported operating system.
Installed com.ibm.cic.agent_1.9.2003.20220917_1018 to the
/opt/IBM/InstallationManager/eclipse directory.
Installed com.ibm.gklm42.db2.lin.ofng_11.5.8.0 to the /opt/IBM/DB2GKLMV42
directory.
Installed com.ibm.websphere.liberty.BASE_22.0.12.20221107_1900 to the
/opt/IBM/WebSphere/Liberty directory.
Installed com.ibm.java.jdk.v8_8.0.7020.20221026_1851 to the
/opt/IBM/WebSphere/Liberty directory.
Installed com.ibm.gklm42.linux_4.2.0.0 to the /opt/IBM/GKLMV42 directory.
CRIMA1137W WARNING: The following packages do not support the 64-bit version of
Installation Manager that you are using: IBM Db2 version 11.5.8.0, IBM Security
Guardium Key Lifecycle Manager version 4.2.0.0. If you continue, you might have
issues with installation and deployment. For information about 64-bit mode support
for a package, see the package documentation.

Explanation: The 64-bit version of Installation Manager checks each package for
64-bit support. If a package does not support the 64-bit version, you receive a
warning.

User Action: Use a 32-bit version of Installation Manager to install the package.
WARNING: Support for using Java SE 7 and Java SE 7.1 with WebSphere Liberty ended.
The Liberty kernel can no longer run with Java SE 7 or Java SE 7.1.
Installation process is complete. Please look into Installation Manager logs for
details.

[root@gklm02 disk1]#
```

Note: Check the kernel parameter *kernel_msgmni* value with command `sysctl -a | grep kernel.msgmni`.

If the value is set to current **RAM * 1024** (in GB), then ignore the warning from the kernel parameters pre-check.

Ignore the error message not supported operating system displayed during the installation. This is an error returned from the install check utility for SGKLM 4.2. RHEL 8.8 is listed as supported. For more information, see [IBM Security Guardium Key Lifecycle Manager Support Matrix](#).

11. Modify Db2 permissions to allow GKLM access to the database on host GKLM02. See Example 2-24.

Example 2-24

```
root@gklm02 [/]# su - klmb42
```

```
[klmb42@gklm02 ~]$ db2 connect to klmb42
```

```
Database Connection Information
```

```
Database server          = DB2/LINUX8664 11.5.8.0
SQL authorization ID     = KLMB42
Local database alias    = KLMB42
```

```
[klmb42@gklm02 ~]$ db2 grant secadm on database to user klmb420
DB20000I The SQL command completed successfully.
[klmb42@gklm02 ~]$
```

12. Backup the current configuration and install the latest FixPack for GKLM on host GKLM02. As of this writing the latest fix pack is version 4.2.0.1. See Example 2-25.

Example 2-25

```
[root@gklm02 [/]# exec bash
```

```
[root@gklm02 [/]# cd /opt/IBM/WebSphere/Liberty/bin/
```

```
[root@gklm02 bin]# ./stopServer.sh
```

```
Stopping server gklm42server.
Server gklm42server stopped.
[root@gklm02 bin]#
```

```
[root@gklm02 bin]# mkdir /tmp/wasbackup
[root@gklm02 bin]# cd /tmp/wasbackup/
```

```
[root@gklm02 wasbackup]# tar -cvf wasbackup.tar /opt/IBM/WebSphere/Liberty/*
tar: Removing leading `/' from member names
/opt/IBM/WebSphere/Liberty/bin/
```

```
...
<Trunkated output>
```

```
...
[root@gklm02 wasbackup]#
```

```
[root@gklm02 wasbackup]# cd /opt/IBM/WebSphere/Liberty/bin/
[root@gklm02 bin]# ./startServer.sh
```

```
Starting server gklm42server.
Server gklm42server started with process ID 17475.
[root@gklm02 bin]#
```

```
[root@gklm02 GKLM420]# cd /root/GKLM420/sklm/
```

```
[root@gklm02 sklm]# cp SKLM_Silent_Update_Linux_Resp.xml
SKLM_Silent_Update_Linux_Resp.xml.original
[root@gklm02 sklm]# vi SKLM_Silent_Update_Linux_Resp.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!--The "acceptLicense" attribute has been deprecated. Use "-acceptLicense"
command line option to accept license agreements.-->
<agent-input clean='true'>
<server>
<repository location='/root/GKLM420/javafp/repository.config'/>
<repository location='/root/GKLM420/wasfp/repository.config'/>
<repository location='/root/GKLM420/sklm/repository.config'/>
</server>
<profile id=' IBM Security Guardium Key Lifecycle Manager v4.2.0' installLocation='/opt/IBM/GKLMV42'>
<data key='eclipseLocation' value='/opt/IBM/GKLMV42'/>
<data key='user.SKLM_ADMIN_USER,com.ibm.gklm42.linux' value='SKLMAdmin'/>
<data key='user.SKLM_ADMIN_PASSWORD,com.ibm.gklm42.linux'
value='1y232N6S6QG3PNdv1NvFbg=='/>
<data key='user.DB_ADMIN_USER,com.ibm.gklm42.linux' value='k1mdb42'/>
<data key='user.DB_ADMIN_PASSWORD,com.ibm.gklm42.linux'
value='1y232N6S6QG3PNdv1NvFbg=='/>
</profile>
<install modify='false'>
<offering id='com.ibm.java.jdk.v8' profile='WebSphere Liberty V_220012'
installFixes='none' version='8.0.8000.20230314_0926'/>
<offering id='com.ibm.websphere.liberty.BASE' profile='WebSphere Liberty V_220012'
installFixes='none' version='23.0.3.20230319_1900'/>
<offering id='com.ibm.gklm42.linux' version='4.2.0000.1' profile='IBM Security
Guardium Key Lifecycle Manager v4.2.0' features='main.feature'
installFixes='none'/>
</install>
</agent-input>
```

13. Save changes to the file SKLM_Silent_Update_Linux_Resp.xml. See Example 2-26.

Example 2-26

```
[root@gklm02 GKLM420]# cd /root/GKLM420/
[root@gklm02 GKLM420]# ./silent_updateSKLM.sh /opt/IBM/InstallationManager/
/opt/IBM/WebSphere/Liberty/
About to install FixPack...
Name: WebSphere Liberty V_220012
Launching InstallManager...
/opt/IBM/InstallationManager//eclipse/tools/imcl -input
/root/GKLM420/sklm/SKLM_Silent_Update_Linux_Resp.xml -silent -acceptLicense
Updated to com.ibm.java.jdk.v8_8.0.8000.20230314_0926 in the
/opt/IBM/WebSphere/Liberty directory.
```

```
Updated to com.ibm.websphere.liberty.BASE_23.0.3.20230319_1900 in the
/opt/IBM/WebSphere/Liberty directory.
Updated to com.ibm.gklm42.linux_4.2.0.1 in the /opt/IBM/GKLMV42 directory.
CRIMA1137W WARNING: The following packages do not support the 64-bit version of
Installation Manager that you are using: IBM Security Guardium Key Lifecycle
Manager version 4.2.0.1, IBM Security Guardium Key Lifecycle Manager installed
version 4.2.0.0. If you continue, you might have issues with installation and
deployment. For information about 64-bit mode support for a package, see the
package documentation.
```

Explanation: The 64-bit version of Installation Manager checks each package for 64-bit support. If a package does not support the 64-bit version, you receive a warning.

User Action: Use a 32-bit version of Installation Manager to install the package.
WARNING: Support for using Java SE 7 and Java SE 7.1 with WebSphere Liberty ended. The Liberty kernel can no longer run with Java SE 7 or Java SE 7.1.
Installation of fix pack is complete. See the Installation Manager log files for more information.
[root@gklm02 GKLM420]#

This completes the installation of the GKLM server, GKLM02.

2.3 GKLM web-based GUI

An IBM GKLM server installation includes a web-based GUI that can be used for the following tasks:

- ▶ reading the current configuration
- ▶ performing a manual backup or restore of the GKLM server configuration and configured keys
- ▶ managing web-based GUI users
- ▶ configuring replication between master and clone servers

The IBM GKLM server web-based GUI is accessible on every GKLM server installation at `https://<GKLM SERVER IP>:9443`.

For this example environment, the following addresses are used:

- ▶ GKLM01: <https://10.134.184.58:9443> (MASTER)
- ▶ GKLM02: <https://10.134.184.48:9443> (CLONE)

To install the GKLM server, the web-based GUI will be used for the following tasks:

- ▶ GKLM Server certificate creation
- ▶ GKLM Server replication setup between configured master and clone servers
- ▶ Creation of GKLM Server web-based GUI users

During the initial installation of the IBM GKLM server instance, a default administrative user account, SKLMAdmin, is automatically created.

The password for user SKLMAdmin is the password that was defined before the installation when the encrypted password string was created. See Example 2-8 on page 11. For the

example in this document, we used the same password for SKLMAdmin on both of the servers, GKLM01 and GKLM02.

To login to IBM Security Guardium Key Lifecycle Manager, open a browser at the IP address of the server.



Figure 2-1 IBM GKLM server login screen

2.3.1 GKLM Server License Activation

After initial installation of IBM Security Guardium Key Lifecycle Manager the server will be running in trial mode until a valid license is applied. Activate both GKLM servers, GKLM01 and GKLM02, with a permanent license from the web-based GUI.

Note: The downloaded and renamed license file `gklm.license.zip` must be available on the workstation where the license installation is being performed.

Install a permanent license to a GKLM server.

1. From the Trial version notification located in the upper right corner of the web-based GUI, select **Apply License** to open the Apply License page. See Figure 2-2 and Figure 2-3 on page 23.

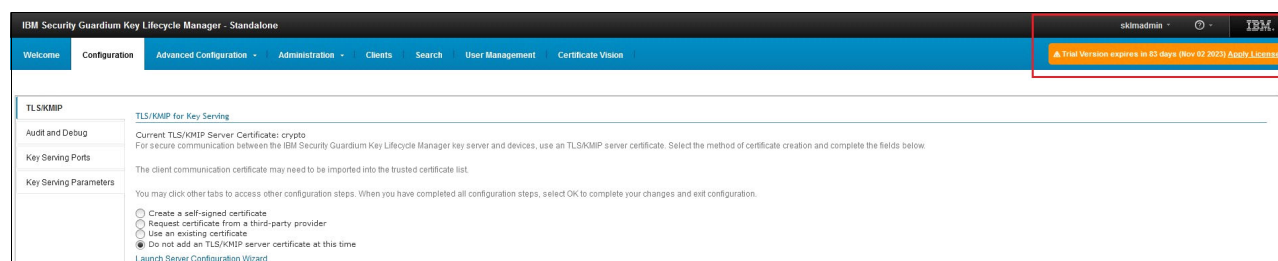
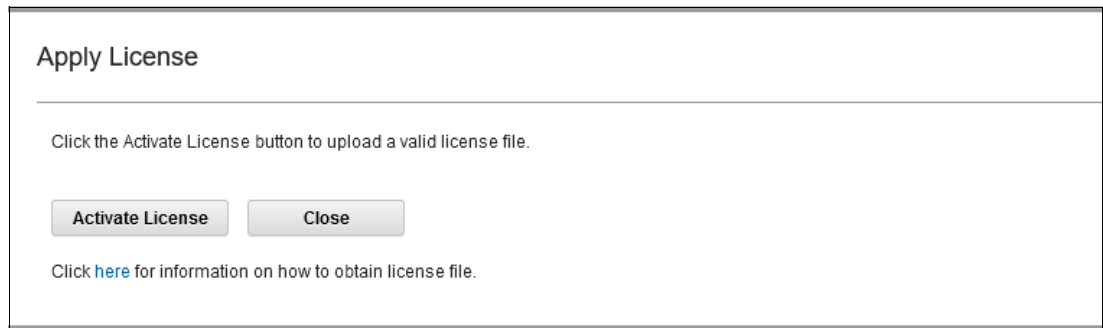


Figure 2-2 Trial version warning message is shown until license is applied



Apply License

Click the Activate License button to upload a valid license file.

Click [here](#) for information on how to obtain license file.

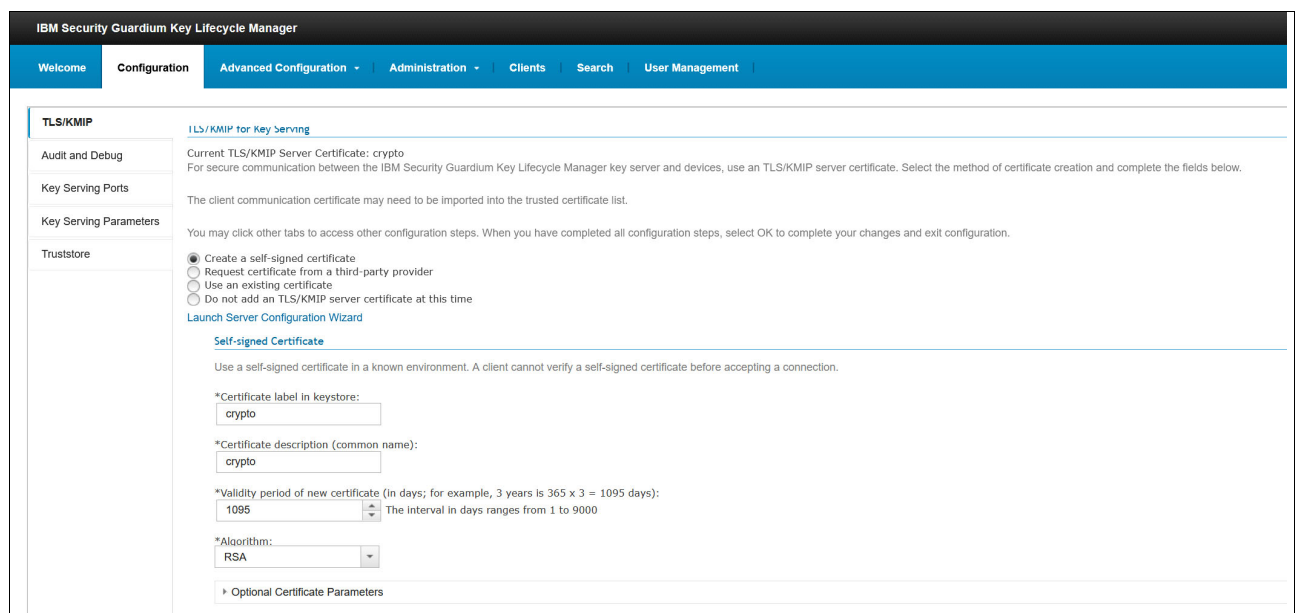
Figure 2-3 Apply License panel

2. Select **Activate License** and locate the license file `gklm.license.zip`.
After the upload of the license file is complete, the server displays a notification License is valid and the user session will be logged out.
3. Login again to the web-based GUI as SKLMAdmin and confirm that the Trial version message is not displayed.

2.3.2 GKLM Server Certificate Creation

Before the GKLM server can be configured to serve encryption keys or for replication, a server certificate appropriate for your environment must be created.

1. On the server that you have selected as the master server, GKLM01 in this example, a server certificate must be created with the settings as shown in Figure 2-4.
 - a. Select **Create a self-signed certificate**.
 - b. Enter a label and description in both the **Certificate label in the keystore** field and the **Certificate description (common name)** field.
 - c. Enter the number of days in the **Validity period of the new certificate** field. In this example, the Algorithm is RSA.



IBM Security Guardium Key Lifecycle Manager

Welcome | Configuration | Advanced Configuration | Administration | Clients | Search | User Management

TLS/KMIP | [TLS/KMIP for Key Serving](#)

Audit and Debug

Key Serving Ports

Key Serving Parameters

Truststore

Current TLS/KMIP Server Certificate: crypto
For secure communication between the IBM Security Guardium Key Lifecycle Manager key server and devices, use an TLS/KMIP server certificate. Select the method of certificate creation and complete the fields below.

The client communication certificate may need to be imported into the trusted certificate list.

You may click other tabs to access other configuration steps. When you have completed all configuration steps, select OK to complete your changes and exit configuration.

☒ Create a self-signed certificate
☐ Request certificate from a third-party provider
☐ Use an existing certificate
☐ Do not add an TLS/KMIP server certificate at this time

[Launch Server Configuration Wizard](#)

Self-signed Certificate

Use a self-signed certificate in a known environment. A client cannot verify a self-signed certificate before accepting a connection.

*Certificate label in keystore:
crypto

*Certificate description (common name):
crypto

*Validity period of new certificate (in days; for example, 3 years is 365 x 3 = 1095 days):
1095 The interval in days ranges from 1 to 9000

*Algorithm:
RSA

Optional Certificate Parameters

Figure 2-4 IBM GKLM server certificate creation settings view

Figure 2-5 shows a summary of the configuration.

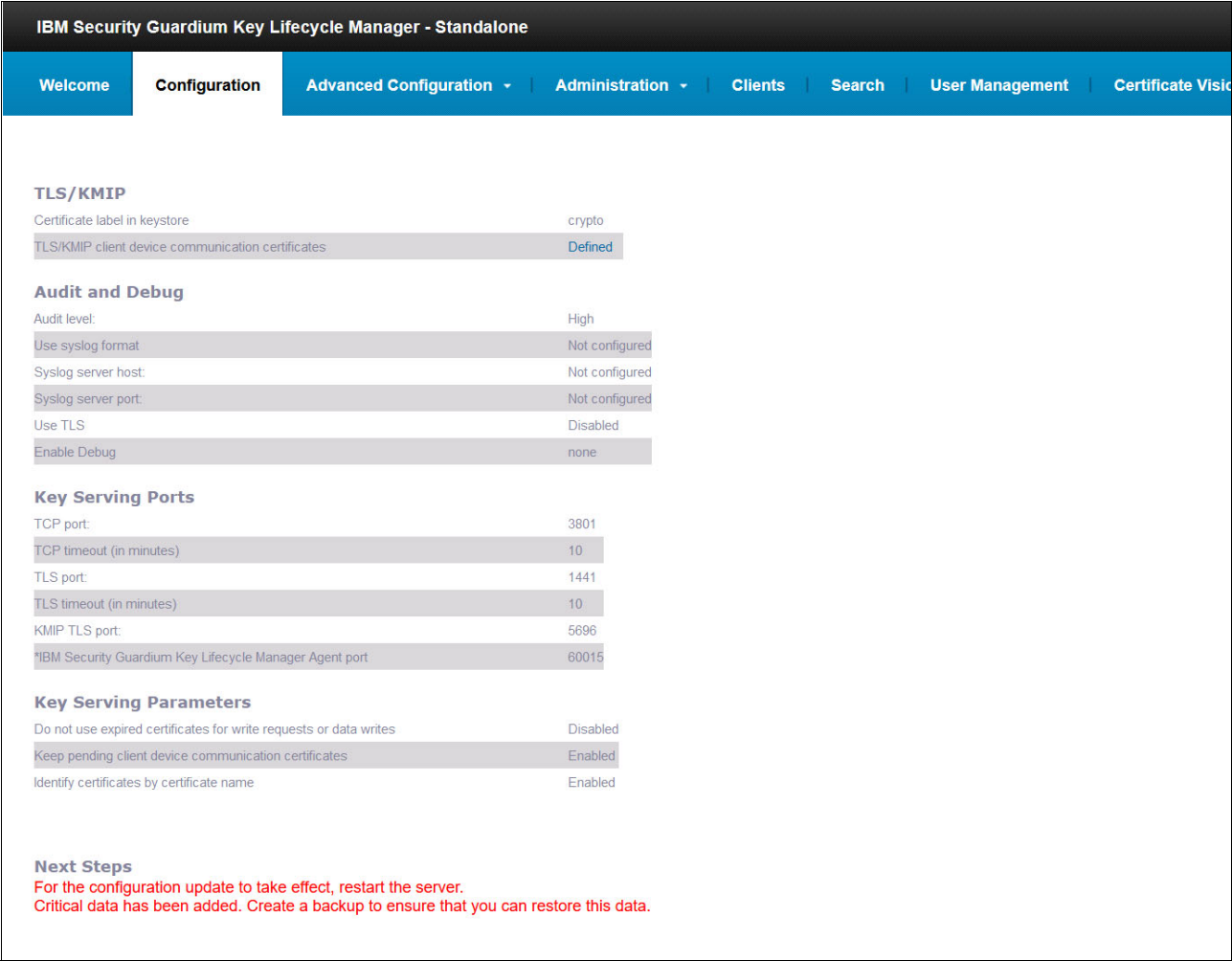


Figure 2-5 IBM GKLM TLS/KMIP configuration confirmation window

- 2. After the server certificate is created, the server must be restarted for the changes to take effect and to apply the certificate.
 - a. Select **skladmin** to open the drop-down menu. Select **Restart Server**.

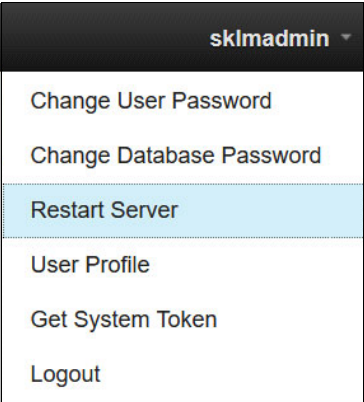


Figure 2-6 Restart Server option

- b. After **Restart Server** is selected, click **OK** to confirm the restart of the server. See Figure 2-7

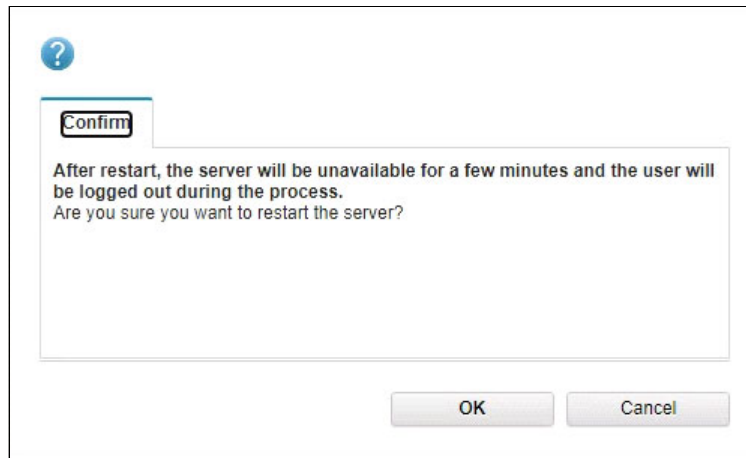


Figure 2-7 Confirm restart action dialog

After the server is restarted, the server certificate that is currently in use can be confirmed from the Configuration section of the GUI.

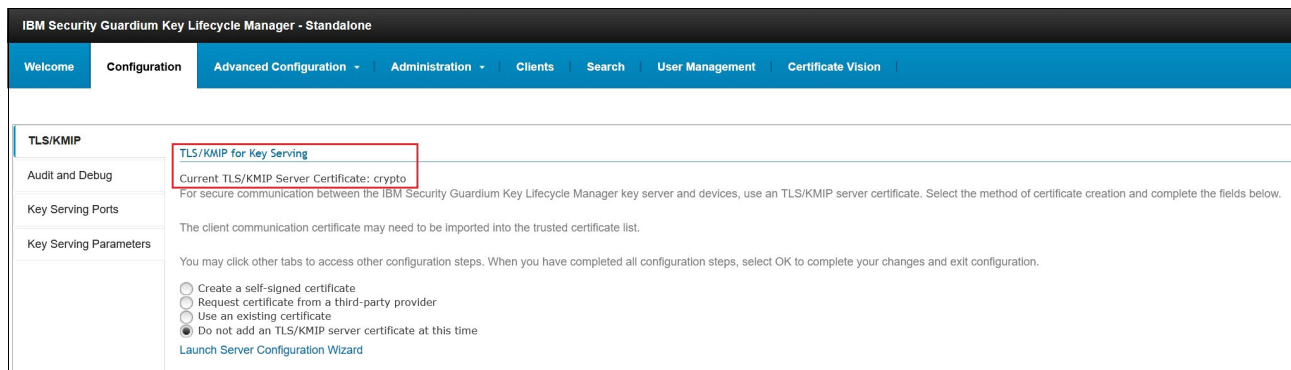


Figure 2-8 Confirming the certificate being used by the GKLM Server

When the correct key name is being served, the configuration is complete.

2.4 GKLM server replication setup

When a malfunction occurs on the current master server and it becomes unavailable, GKLM allows for increased availability and faster recovery of encryption keys by configuring a replication server. The replication server is then able to provide the client nodes with the needed encryption keys. In this example we configure replication between GKLM servers GKLM01 and GKLM02.

After replication is configured, the following data on the master server is copied to the replication server:

- ▶ IBM Security Guardium Key Lifecycle Manager database tables
- ▶ Truststore and keystore with the master key
- ▶ IBM Security Guardium Key Lifecycle Manager configuration files

2.4.1 Backing up and copying the master server configuration

Create a backup of the master server (GKLM01) configuration and its data that can then be restored on the clone server (GKLM02) then configure replication. To do this, select **Administration** → **Backup and Restore** on the web-based GUI of the GKLM01 server. The backup can also be restored on any other server that is running the same version of the IBM Security GKLM application. The backup can be used if both configured production servers GKLM01 and GKLM02 are unavailable.

To restore configuration on any server, the password of the backup file must be known.

1. From the Administration drop-down menu, select **Backup and Restore**. See Figure 2-9.

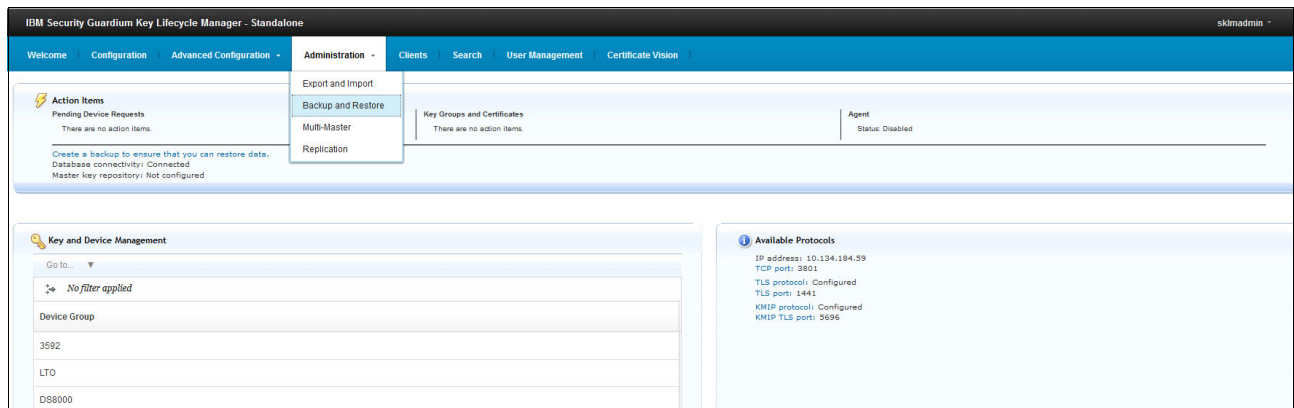


Figure 2-9 Backup and Restore window in GKLM GUI

2. Select **Create** to create a new backup. See Figure 2-10.

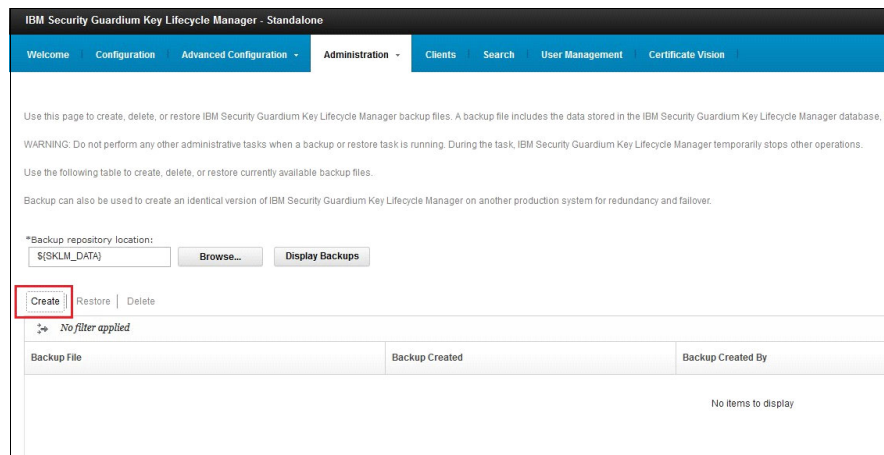


Figure 2-10 Create button in the GKLM web-based GUI

3. Enter a password and, if wanted, a description into the Create Backup panel, then click **Create Backup**. See Figure 2-11 on page 27.

Note: Ensure that the password is stored or recorded safely for any backup file created. This password is required to be entered during the restore of this backup file.

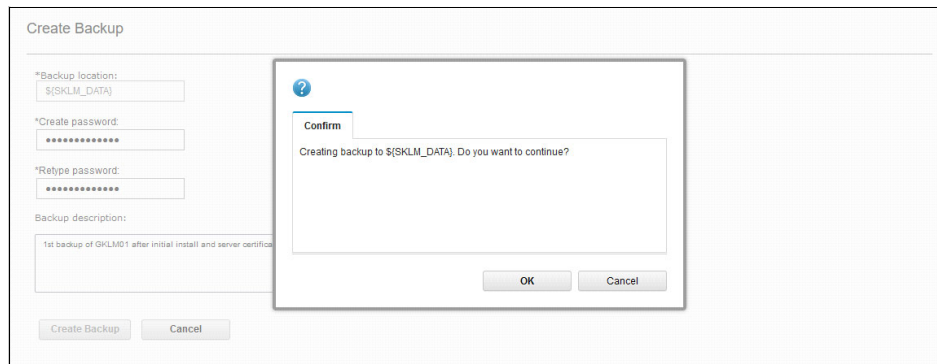


Figure 2-11 Create Backup options and backup creation dialog confirmation

After the backup operation is completed, the backup file name and location is displayed in the list of currently available backups.

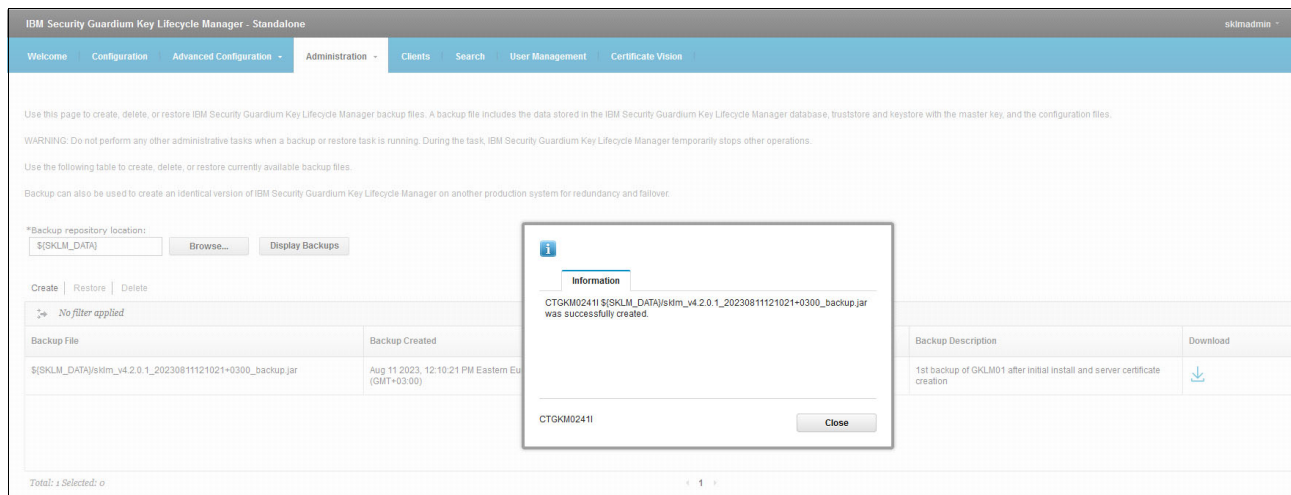


Figure 2-12 Successful backup details information window

- To prepare to create a clone of the master server, transfer the backup file from the master server GKLM01 to the clone server GKLM02. See Example 2-27. The example uses **scp** to transfer the backup file.

Example 2-27 Transferring master server backup to clone server

```
[root@gklm01 /]# ls -la /opt/IBM/WebSphere/Liberty/products/sklm/data/
total 68
drwxr-x--- 4 klmdb42 root    86 Aug 11 12:10 .
drwxr-x--- 15 klmdb42 root 4096 Aug  4 12:29 ..
drwxr-x--- 2 klmdb42 root    6 Aug  4 12:27 agent
drwxr-x--- 2 klmdb42 root    6 Aug  4 12:27 restore
-rw-rw---- 1 klmdb42 klmdb42 64836 Aug 11 12:10 sklm_v4.2.0.1_20230811121021+0300_backup.jar
[root@gklm01 /]#
[root@gklm01 /]# scp
/opt/IBM/WebSphere/Liberty/products/sklm/data/sklm_v4.2.0.1_20230811121021+0300_backup.jar
root@gklm02:/opt/IBM/WebSphere/Liberty/products/sklm/data
root@gklm02's password:
sklm_v4.2.0.1_20230811121021+0300_backup.jar 100% 63KB
26.6MB/s 00:00
[root@gklm01 /]#
```

2.4.2 Restoring master server configuration to the clone server

The next step in creating a replication pair is to restore the configuration of the master server to the clone server that will receive the replicated data.

1. On the clone server GKLM02, move or copy the backup file to the restore folder and confirm.
2. Use the **chown** command to assign the file owner and group to `klmdb42:klmdb42`. See Example 2-28.

Example 2-28 Copying GKLM server backup to restore location on the clone server

```
[root@gklm02 /]# cp
/opt/IBM/WebSphere/Liberty/products/sklm/data/sklm_v4.2.0.1_20230811121021+0300_backup.jar
/opt/IBM/WebSphere/Liberty/products/sklm/data/restore/

[root@gklm02 /]# chown klmdb42:klmdb42
/opt/IBM/WebSphere/Liberty/products/sklm/data/restore/sklm_v4.2.0.1_20230811121021+0300_backup.jar

[root@gklm02 /]# ls -la /opt/IBM/WebSphere/Liberty/products/sklm/data/restore/
total 64
drwxr-x--- 2 klmdb42 root    58 Aug 11 14:06 .
drwxr-x--- 4 klmdb42 root    86 Aug 11 14:03 ..
-rw-r----- 1 klmdb42 klmdb42 64836 Aug 11 14:06 sklm_v4.2.0.1_20230811121021+0300_backup.jar
[root@gklm02 /]#
```

Note: If the owner of the backup file is not set to `klmdb42:klmdb42` on the target system, the restore is not performed and the backup file is not visible on the web-based GUI.

3. Log in into the clone server GKLM02 web-based GUI and select **Administration** → **Backup&Restore**.
4. After the window opens, click **Browse**, if necessary, to select the restore folder. See Figure 2-13.
5. Click **Display Backups** to refresh the contents of that folder.

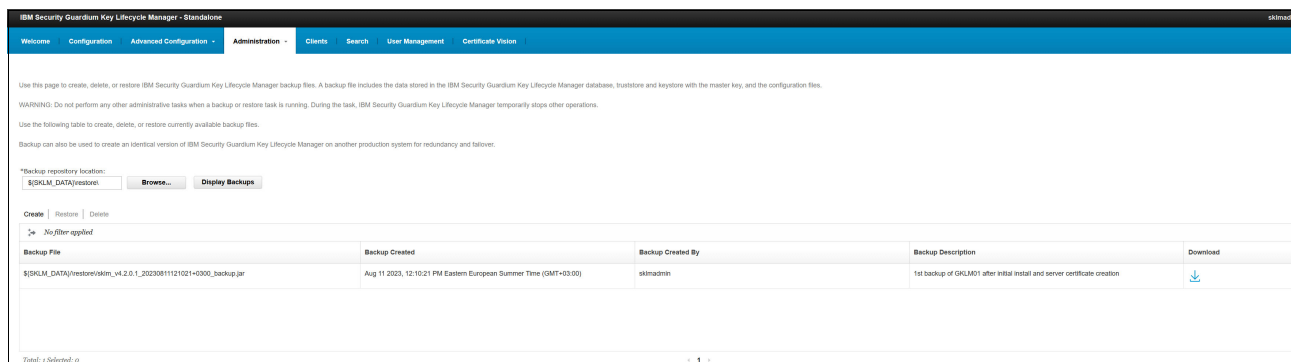


Figure 2-13 Displaying backup versions available for the GKLM server

6. Select a file from the list of available backup files and select **Restore**.
7. On the Restore from Backup window, provide the password that is associated with the backup file. Click **Restore Backup**. See Figure 2-14 on page 29.

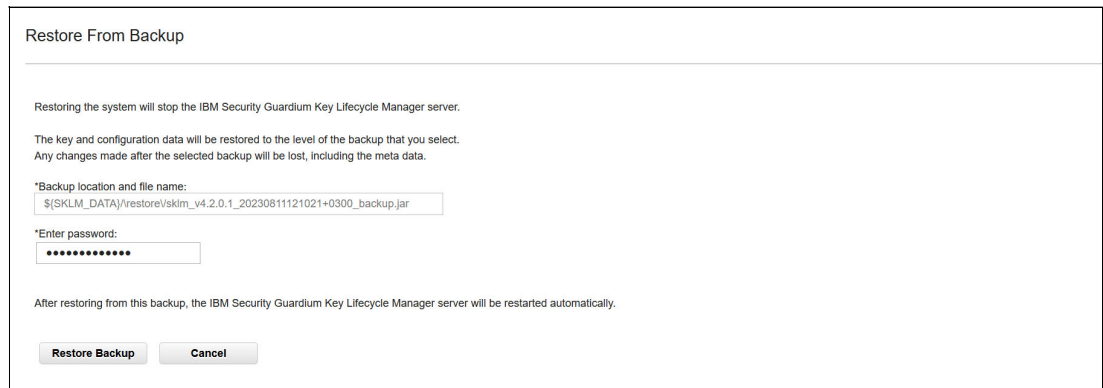


Figure 2-14 Restore from Backup dialog window

8. Click **OK** to start the restore operation. After the restore is completed, the server restarts. See Figure 2-15.

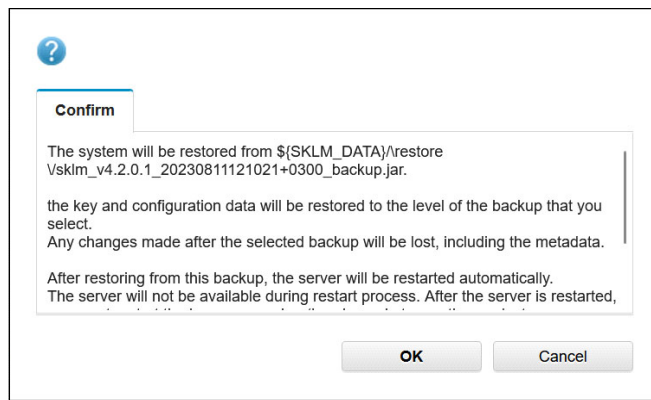


Figure 2-15 Restore backup version confirmation

After the restore is complete, the system the server displays a message that the restore was successful. and then the IBM Security Guardium Key Lifecycle Manager application restarts. See Figure 2-16 on page 30.

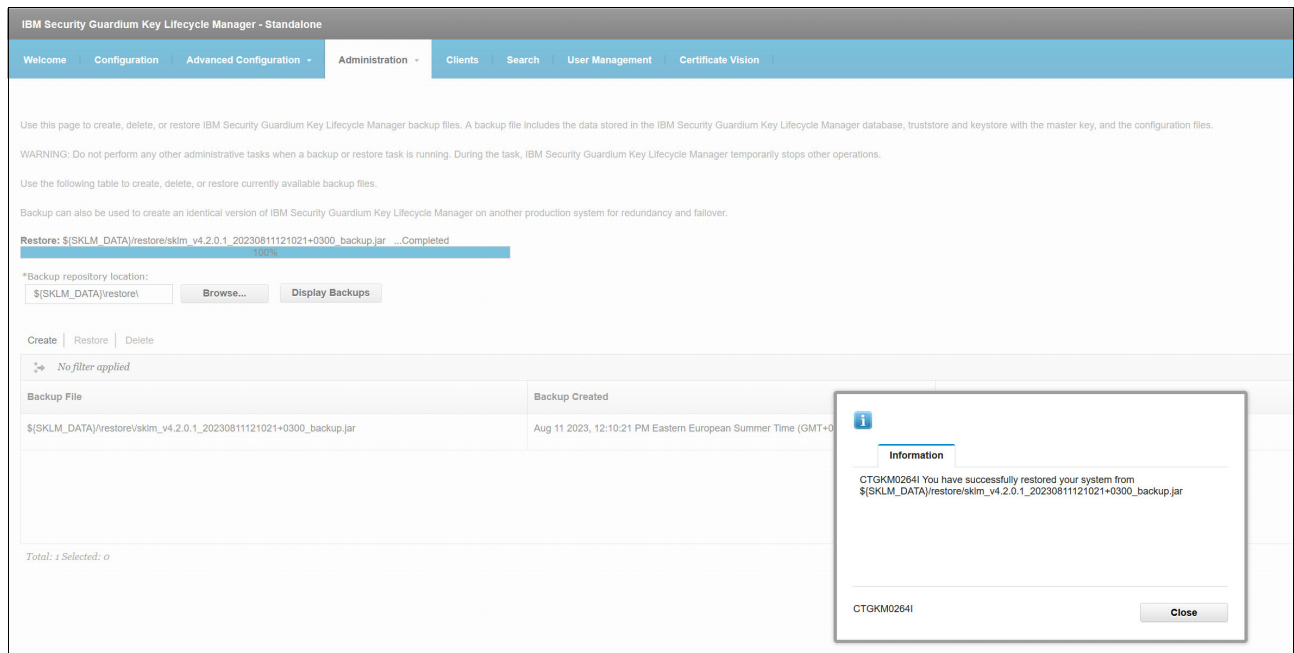


Figure 2-16 Successful restore confirmation message

2.4.3 Configuring replication on the master server

1. From the web-based GUI of the master server GKLM01 select **Administration -> Replication**.
2. Select **Master** to assign the master server GKLM01. See Figure 2-17.
3. Click **OK** to reset the server configuration.

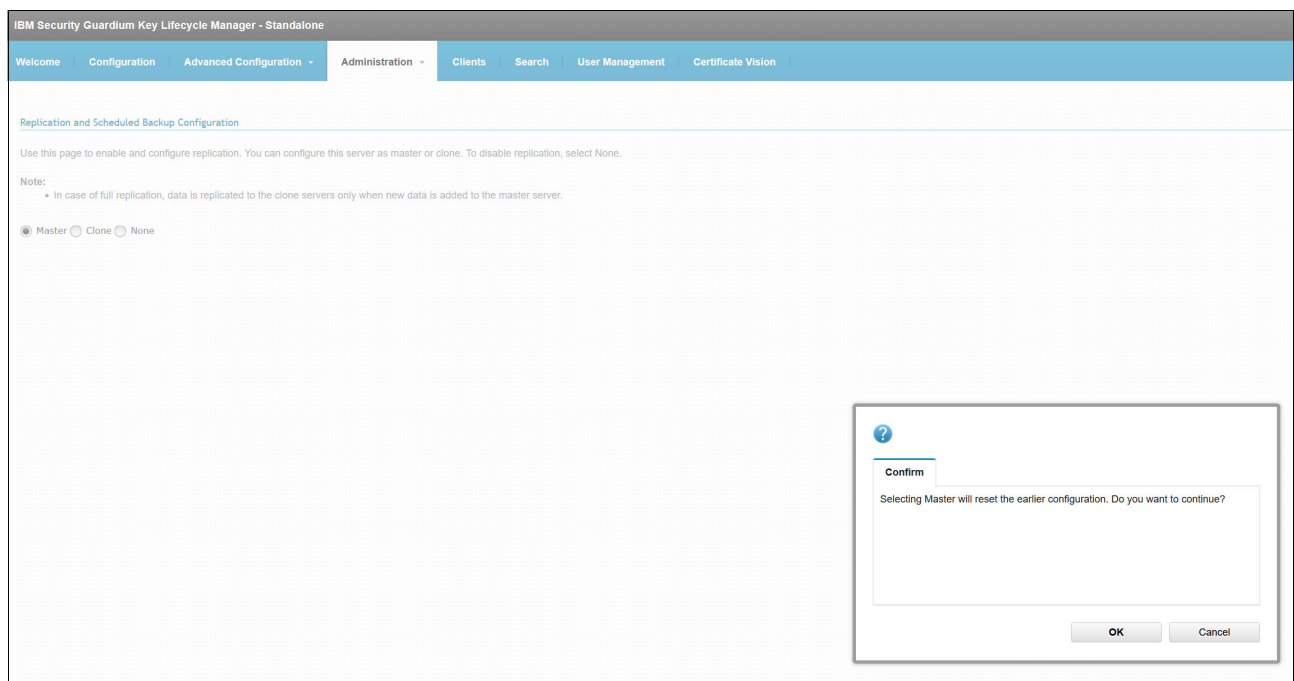


Figure 2-17 Confirmation when resetting server configuration

4. In the Replication and Scheduled Backup Configuration window, complete the fields in the Basic Properties tab. See Figure 2-18.

For the example configuration, the following settings are used in each field:

- Certificate: crypto
- Replication backup encryption password: P44kaupunk1_Turku
- Master listen port: 1111
- Clone server: gklm02.cloud.stg.forum.ibm.com
- Clone listen port: 2222

IBM Security Guardium Key Lifecycle Manager - Standalone

Welcome | Configuration | Advanced Configuration | Administration | Clients | Search | User Management | Certificate Vision

Replication and Scheduled Backup Configuration

Use this page to enable and configure replication. You can configure this server as master or clone. To disable replication, select None.

Note:

- In case of full replication, data is replicated to the clone servers only when new data is added to the master server.

☒ Master ☐ Clone ☐ None

[Start Replication Server](#) [Replicate Now](#)

Basic Properties | Advanced Properties

*Certificate from keystore:

*Replication backup encryption passphrase:

*Confirm replication backup encryption passphrase:

*Master listen port:

▼ Clone Details

[Add Clone](#)

Clone-1	IP Address or Host Name:	Clone-1	Port:	
	<input type="text" value="gklm02.cloud.stg.forum.ibm.cc"/>		<input type="text" value="2222"/>	Delete

[OK](#)

Figure 2-18 Master server properties configuration view

Note: Do *not* change the settings in the Advanced Properties tab.

5. After entering the values, click **OK**.

6. When prompted, click **OK** to save the replication configuration. See Figure 2-19.

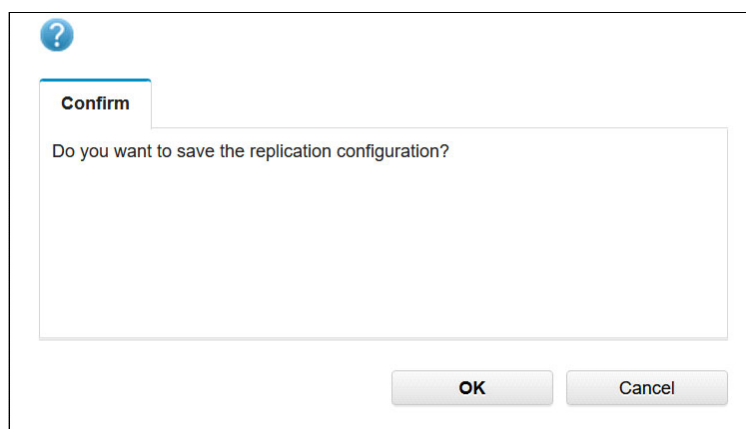


Figure 2-19 Confirm replication configuration change window

2.4.4 Configuring replication on the CLONE server

1. From the web-based GUI of the clone server GKLM02, select **Administration** → **Replication** then select the **Clone** option. See Figure 2-20.
- Confirm the port numbers:
 - Master listen port: 1111
 - Clone listen port: 2222

Note: Leave Advanced properties set to the default settings.

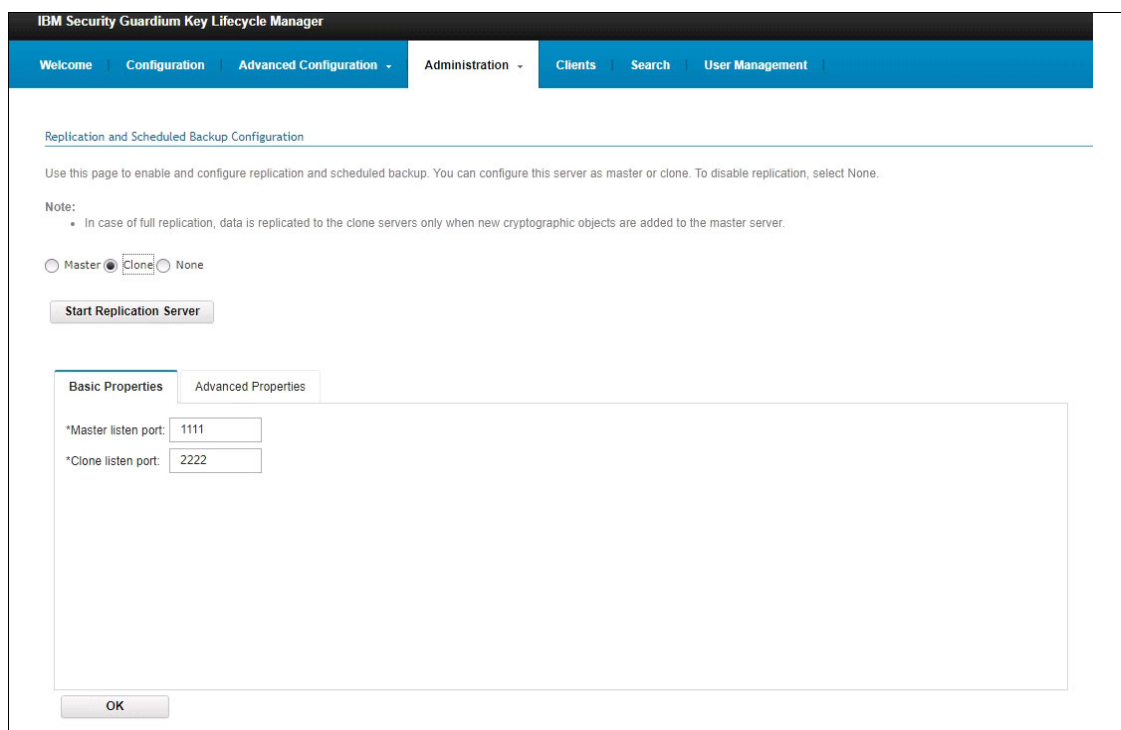


Figure 2-20 Configure Clone GKLM server settings

2. Click **OK** to save the configuration. Wait for the save to finish.
3. After the configuration is saved, select **Start Replication Server** and confirm that no errors are reported.

Note: Replication must be started on both the master and clone servers.

2.4.5 Confirming replication is working

When replication has been configured and enabled on both the master and clone servers, test the replication from the Master server GKLM01.

From the web-based GUI, select **Administration** → **Replication**.

Test replication by clicking **Replicate Now** and confirm that replication was successful.

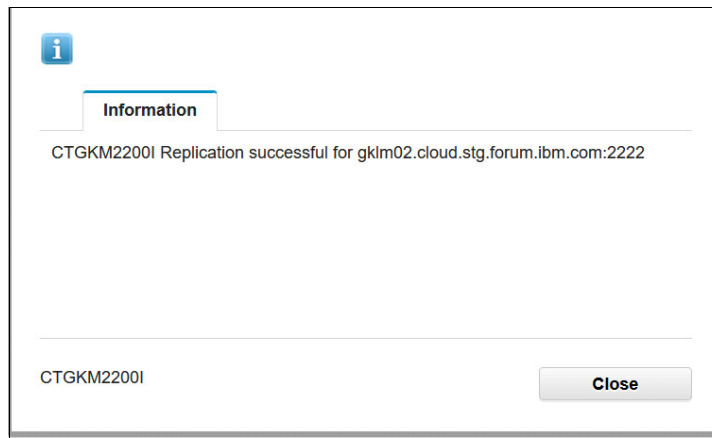


Figure 2-21 Replication status confirmation message

Note: More details related to replication events can be found by reviewing the logs on both the master and clone servers. Replication Log files are located in `/opt/IBM/WebSphere/Liberty/products/sklm/logs/replication` on each server.

2.5 GKLM Server Users

The GKLM server web-based GUI has a default Administrator user `sklmAdmin`. The password for `sklmAdmin` is set during the initial installation of the GKLM server instance.

In addition to the default Administrator user, other user IDs can be configured from the GKLM server web-based GUI. Each user can be a member of a predefined or a user-created role and group, depending on the role of the user. See Figure 2-22 on page 34.

Users		
Add Modify Delete		
No filter applied		
User Name	Assigned Roles	Assigned Groups
luis (luis)	kimSecurityOfficer, GPFS, kimClientUser	kimGUIAccessGroup, kimSecurityOfficerGroup, kimBackupRestoreGroup
mika (mika)	kimSecurityOfficer, XIV, kimClientUser	kimGUIAccessGroup, kimSecurityOfficerGroup, kimBackupRestoreGroup
skimadmin (skimadmin)	kimSecurityOfficer, kimClientUser	kimGUIAccessGroup, kimSecurityOfficerGroup
turku (turku)	kimClientUser	kimSecurityOfficerGroup

Figure 2-22 GLKM Server GUI Users panel

Note: Users that are configured locally on a GKLM server instance are *not* replicated between a Master and Clone server by GKLM replication. The administrator that defines GKLM users must maintain users manually on each server.

2.6 GKLM Servers and Clients

In this example, in the current environment only GKLM servers are installed and configured. Next, add the GKLM clients to the example environment. For more information, see chapter 3 “IBM Storage Scale File System and encryption configuration”.

Table 2-1 shows the configuration information for the current GKLM servers. See Table 2-2 on page 35 for a list of GKLM clients to be added.

Table 2-1 GKLM server details

Name	IP address	GKLM Version	Role
GKLM01	10.134.184.59	4.2.0.1	Master
GKLM02	10.134.184.48	4.2.0.1	Clone

GKLM servers include the following components, which are installed and configured at initial installation time:

- ▶ IBM Security Key Lifecycle Manager
- ▶ IBM WebSphere® Application Server
- ▶ IBM Db2 relational database

IBM Security Key Lifecycle Manager stores its key materials in a Db2 relational database.

Manual backups of the GKLM server configuration and its stored keys can be performed from the web-based GUI of the server at any time by selecting **Administration** → **Backup&Restore** then clicking **Create**. See Figure 2-23 on page 35.

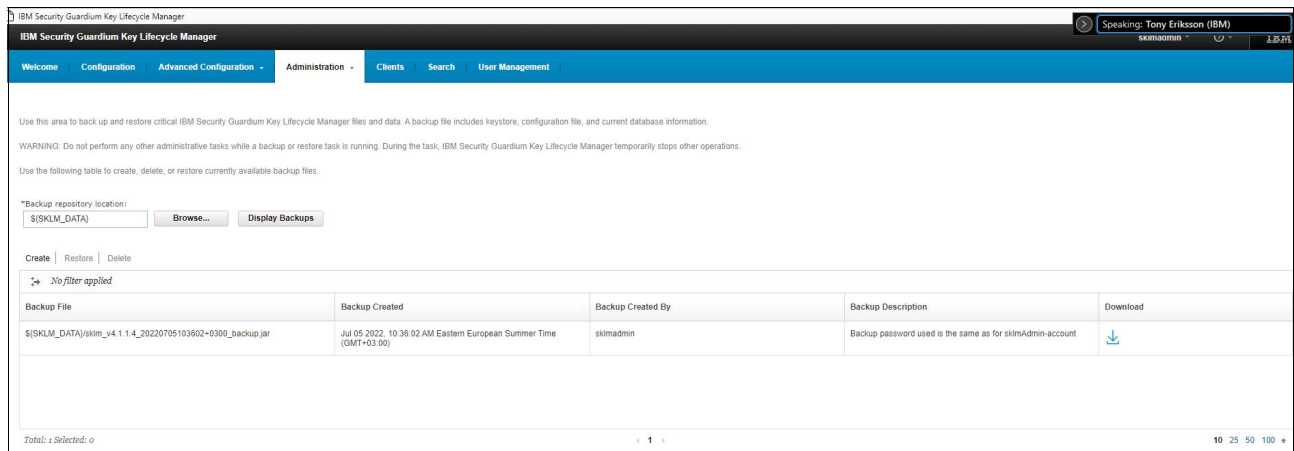


Figure 2-23 GKLM server administration window backup file list

The GKLM server automatically backs up its configurations and stored keys every 24 hours. Automatic backup of the configuration also occurs if new encryption keys are created or if the configuration of the GKLM server is changed.

Manual and automatic backups are stored by default on a local file system of the GKLM server. The backup files should be backed up periodically by an external backup procedure.

Clients

Each GKLM client is an IBM Storage Scale cluster. Table 2-2 lists all clusters currently configured with GKLM-based encryption and their member nodes.

Table 2-2 Client configuration details

Name	IP address	IBM Storage Scale Version	Cluster
Scale1	10.134.184.51	DME / 5.1.7.1	RB_Storage_Cluster.cloud.stg.forum.ibm.com
Scale2	10.134.184.52	DME / 5.1.7.1	RB_Storage_Cluster.cloud.stg.forum.ibm.com
Scale3	10.134.184.53	DME / 5.1.7.1	RB_Storage_Cluster.cloud.stg.forum.ibm.com
RemoteScale1	10.134.184.54	DME / 5.1.8.1	RB_Remote_Cluster.cloud.stg.forum.ibm.com
RemoteScale2	10.134.184.55	DME / 5.1.8.1	RB_Remote_Cluster.cloud.stg.forum.ibm.com
RemoteScale3	10.134.184.56	DME / 5.1.8.1	RB_Remote_Cluster.cloud.stg.forum.ibm.com
Moomi	10.134.184.47	DME / 5.1.8.1	RB_Muumilaakso.cloud.stg.forum.ibm.com



IBM Storage Scale file system and encryption configuration

This chapter includes a step-by-step procedure to set up an environment for an encrypted file system. This includes instructions to set up and configure IBM Storage Scale clients and file systems to encrypt their data.

Installation and initial configuration of IBM Storage Scale client nodes and clusters is not covered in this publication.

The following topics are presented in this chapter:

- ▶ 3.1, “IBM Storage Scale clients” on page 38
- ▶ 3.2, “GKLM Server configuration” on page 39
- ▶ 3.3, “IBM Storage Scale configuration” on page 41
- ▶ 3.4, “Configuring Remote clusters and remote cluster mounts” on page 50
- ▶ 3.5, “Enabling encryption on selected file systems and filesets” on page 54
- ▶ 3.6, “File system and file access and attributes of encrypted files” on page 63

3.1 IBM Storage Scale clients

When you configure a Guardium Key Lifecycle Manager (GKLM) server environment, IBM Storage Scale Systems client nodes are responsible for the encryption and decryption of the accessed data. There is no performance impact on the storage cluster nodes or other clients that are accessing non-encrypted data on a storage cluster.

To demonstrate how to take advantage of GKLM-based encryption in an IBM Storage Scale environment, the following example is configured with 3 IBM Storage Scale clusters running on RHEL 8.8. These clusters are described in Table 3-1.

In this publication, the examples include clusters with names RB_Storage_Cluster, RB_Remote_Cluster, and RB_Muumilaakso.

IBM Storage Scale cluster RB_Storage_Cluster has storage devices that are attached to it and has a local IBM Storage Scale file system defined.

IBM Storage Scale cluster RB_Remote_Cluster does not have any local IBM Storage Scale file systems and will be performing remote cluster mounts from RB_Storage_Cluster.

Both IBM Storage Scale clusters will be defined as clients of the GKLM servers.

Environment details for the IBM Storage Scale nodes and clusters are provided in Table 3-1.

Table 3-1 IBM Storage Scale nodes and cluster example settings

Name	IP address	Storage Scale version	Cluster	Has local storage
Scale1	10.134.184.51	DME / 5.1.7.1	RB_Storage_Cluster.cloud.stg.forum.ibm.com	Yes
Scale2	10.134.184.52	DME / 5.1.7.1	RB_Storage_Cluster.cloud.stg.forum.ibm.com	Yes
Scale3	10.134.184.53	DME / 5.1.7.1	RB_Storage_Cluster.cloud.stg.forum.ibm.com	Yes
RemoteScale1	10.134.184.54	DME / 5.1.8.1	RB_Remote_Cluster.cloud.stg.forum.ibm.com	No
RemoteScale2	10.134.184.55	DME / 5.1.8.1	RB_Remote_Cluster.cloud.stg.forum.ibm.com	No
RemoteScale3	10.134.184.56	DME / 5.1.8.1	RB_Remote_Cluster.cloud.stg.forum.ibm.com	No
Moomi	10.134.184.47	DME / 5.1.8.1	RB_Muumilaakso.cloud.stg.forum.ibm.com	No

In this example environment, the encryption is implemented with file system policy per fileset. Access to data is granted for the clients at IBM Storage Scale cluster level per each encrypted fileset.

We create 5 different independent filesets inside file system fs01 and enable encryption for these filesets as shown in Table 3-2 on page 39.

Filesets Turku and Järvenpää will be encrypted with one encryption key, filesets Tokyo and New_York will be encrypted with another encryption key and fileset Muumitalo will not be encrypted.

Table 3-2 Client example configuration settings

File system	Fileset	Data Encrypted
fs01		No
fs01	Turku	Yes
fs01	Järvenpää	Yes
fs01	Tokyo	Yes
fs01	New_York	Yes
fs01	Muumitalo	No

3.2 GKLM Server configuration

When you configure an environment, examine the documentation to see which key servers are supported. This example uses IBM GKLM as the key server for IBM Storage Scale clients. Ensure that the IBM Storage Scale System and GKLM server encryption configuration parameters are configured the same.

3.2.1 Configuration steps for the example environment

Check the status of environment variables and change them when necessary.

Checking the cluster parameters

Check the current setting of parameters *FIPS* and *nistCompliance* on all IBM Storage Scale clusters. See Example 3-1 and Example 3-2.

Example 3-1 Checking the IBM Storage Scale cluster *RB_Storage_Cluster* configuration

```
[root@scale1 ~]# mmlsconfig FIPS1402mode
FIPS1402mode no
[root@scale1 ~]#
[root@RemoteScale1 ~]# mmlsconfig nistCompliance
nistCompliance SP800-131A
[root@RemoteScale1 ~]#
```

Example 3-2 Checking the IBM Storage Scale cluster *RB_Remote_Cluster* current configuration.

```
[root@RemoteScale1 ~]# mmlsconfig FIPS1402mode
FIPS1402mode no
[root@RemoteScale1 ~]#
[root@scale1 ~]# mmlsconfig nistCompliance
nistCompliance SP800-131A
[root@scale1 ~]#
```

Checking the server parameters

Examine the *FIPS* and *nistCompliance* parameters on servers gklm01 and gklm02. FIPS requires TLS version 1.2 or later. See Example 3-3.

Example 3-3 Viewing the configuration on gklm01

```
[root@gklm01 /]# less
/opt/IBM/WebSphere/Liberty/products/sklm/config/SKLMConfig.properties
#Fri Aug 11 12:10:24 EEST 2023
cert.validate=false
Audit.handler.file.size=100000
stopAgentInvocation=true
Audit.handler.file.name=logs/audit/sklm_audit.log
TransportListener.ssl.port=1441
KMIPLListener.ssl.port=5696
Audit.syslog.isSSL=false
Transport.ssl.vulnerableciphers.patterns=_RC4_,RSA_EXPORT,_DES_,EDH-RSA-DES-CBC3-S
HA,ECDHE-RSA-DES-CBC3-SHA
AgentListener.ssl.port=60015
enableClientCertPush=true
TransportListener.tcp.port=3801
OpaqueDataProtected=true
Audit.syslog.server.host=
Audit.eventQueue.max=0
Audit.event.outcome=success,failure
backup.keycert.before.serving=false
TransportListener.tcp.timeout=10
config.keystore.name=defaultKeyStore
Audit.isSyslog=false
useSKIDefaultLabels=false
debug=SEVERE
tklm.lockout.attempts=3
enableKeyRelease=false
tklm.encryption.keysize=256
Audit.event.types=all
TransportListener.ssl.clientauthentication=2
key.cert.fileuploadsize=10KB
TransportListener.ssl.timeout=10
enableServedDataArchive=true
TransportListener.ssl.protocols=TLSv1.2
tklm.ts.password=<encrypted_password>
config.hash.algo=SHA-1
displaySecretTags=true
config.keystore.ssl.certalias=crypto
user.gui.init.config=true
tklm.lockout.enable=true
maximum.keycert.expiration.period.in.years=50
isDeleteModifyRestricted=false
tklm.backup.dir=/opt/IBM/WebSphere/Liberty/products/sklm/data
config.keystore.batchUpdateSize=10000
config.keystore.batchUpdateTimer=60000
tklm.encryption.password=<encrypted_password>
backup.export.fileuploadsize=10MB
Audit.syslog.server.port=
fips=off
```

Changing the cluster parameters

You can change the IBM Storage Scale configuration parameters *FIPS* and *nistCompliance* by using the command **mmchconfig**. The changes can be made without a need to restart IBM Storage Scale processes. At the time of writing, IBM Storage Scale supports only **SP800-131A** for parameter *nistCompliance*.

Example 3-4 Changing FIPS and nistCompliance settings

```
mmchconfig FIPS1402mode=no
mmchconfig nistCompliance=SP800-131A
```

Changing the server parameters

To change *FIPS* and *nistCompliance* equivalent configuration parameters on GKLM servers, modify the configuration file at the default location
`/opt/IBM/WebSphere/Liberty/products/sklm/config/SKLMConfig.properties` on each GKLM server.

GKLM server processes must be restarted for configuration changes to take effect.

3.3 IBM Storage Scale configuration

On each IBM Storage Scale cluster that is accessing the encrypted file systems on `RB_Storage_Cluster`, follow the steps outlined in this section.

3.3.1 `RB_Storage_Cluster.cloud.stg.forum.ibm.com` cluster

Add the GKLM servers GKLM01 and GKLM02 to encryption configuration on cluster `RB_Storage_Cluster`.

Perform the following commands from any of the nodes in the cluster.

1. Add key servers to Storage Scale cluster `RB_Storage_Cluster` by using the **mmkeyserv** command. See Example 3-5.

Example 3-5 Adding key servers to Storage Scale cluster `RB_Storage_Cluster`

```
[root@scale1 ~]# mmkeyserv server add gklm01.cloud.stg.forum.ibm.com --backup
gklm02.cloud.stg.forum.ibm.com
Enter password for the key server gklm01.cloud.stg.forum.ibm.com:
The security certificate(s) from gklm01.cloud.stg.forum.ibm.com must be accepted
to continue. View the certificate(s) to determine whether you want to trust the
certifying authority.
Do you want to view or trust the certificate(s)? (view/yes/no) view

Serial number:          341fb4ab
SHA-256 digest:
6029ed54a62f10bd85671aa60f72b6bc91859229cea52481566814e59c62a8bd
Signature:
3bf06cc609e8d57de352a496346572096fb230d9a0b9c209154957f03484715f14157fe3104214f7c0
721f412c8f7ad093f32ed23ba8034b02473576da944d22799c1978fdae28729210faee97687b02ce6f
afe666b77d57b7f4e69eae816b40c9fa3f5af3650af63273ad5f4f827b80ac76a3cc1b089b9c3c8748
ba595f872a96208647a1e44467e9a256910abe50d0eb159461e7f580a44c74b6dc785ada7670e6f3cc
a91202090d3edd9253192f56c18f0e99e49c4e24976954f9112238e1cf37370ee4ecc1ef72045f8371
```

```

9cff3b07d094da16aa0be5ec9f13b86f834f76dad17e3564261e1e5e463dfcb4e4f757717574cb1e2c
a87ed0ef8e018986975b
Signature algorithm:  SHA256WithRSASignature
Key size:              2048
Issuer:               C=us, O=ibm, OU=gklm42server, CN=localhost
Subject:              C=us, O=ibm, OU=gklm42server, CN=localhost
Valid from:           Aug 04 12:28:52 2023 EEST (+0300)
Valid until:          Aug 03 12:28:52 2024 EEST (+0300)

```

```

Do you trust the certificate(s) above? (yes/no) yes
mmkeyserv: Propagating the cluster configuration data to all
            affected nodes. This is an asynchronous process.
[root@scale1 ~]#

```

2. Verify the keyserver configuration on RB_Storage_Cluster. See Example 3-6.

Example 3-6 Verifying keyserver configuration

```

[root@scale1 ~]# mmkeyserv server show
gklm01.cloud.stg.forum.ibm.com
      Type:              ISKLM
      IPA:               10.134.184.59
      User ID:           SKLMAdmin
      REST port:         9443
      Label:             1_gklm01
      NIST:              on
      FIPS1402:          off
      Backup Key Servers: gklm02.cloud.stg.forum.ibm.com
      Distribute:        yes
      Retrieval Timeout: 60
      Retrieval Retry:   3
      Retrieval Interval: 10000
      REST Certificate Expiration: 2024-08-03 12:28:52 (+0300)
      KMIP Certificate Expiration:

```

[root@scale1 ~]#

3. Create a Tenant (Device Group) RBStorageCluster to server GKLM01. See Example 3-7.

Example 3-7 Creating a Tenant (Device Group)

```

[root@scale1 ~]# mmkeyserv tenant add RBStorageCluster --server
gklm01.cloud.stg.forum.ibm.com
Enter password for the key server gklm01.cloud.stg.forum.ibm.com:
mmkeyserv: Propagating the cluster configuration data to all
            affected nodes. This is an asynchronous process.
[root@scale1 ~]#

```

4. Register IBM Storage Scale cluster RB_Storage_Cluster as a client on server GKLM01 with name RBClientCluster1. See Example 3-8.

Example 3-8 Registering IBM Storage Scale cluster RB_Storage_Cluster as a client

```

[root@scale1 ~]# mmkeyserv client create RBClientCluster1 --server
gklm01.cloud.stg.forum.ibm.com
Enter password for the key server gklm01.cloud.stg.forum.ibm.com:

```

Create a pass phrase for keystore: **<securely store this password. It is needed later>**

Confirm your pass phrase: **<securely store this password. It is needed later>**

mmkeyserv: Propagating the cluster configuration data to all affected nodes. This is an asynchronous process.

[root@scale1 ~]#

5. Register client RBClientCluster1 to a tenant RBStorageCluster. See Example 3-9.

Example 3-9 Registering client RBClientCluster1

```
[root@scale1 ~]# mmkeyserv client register RBClientCluster1 --tenant
RBStorageCluster --rkm-id Turku_Jarvenpaa
Enter password for the key server of client RBClientCluster1:
mmkeyserv: [I] Client currently does not have access to the key. Continue the
registration process ...
mmkeyserv: Successfully accepted client certificate
mmkeyserv: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
[root@scale1 ~]#
```

6. Create a key for cluster RB_Storage_Cluster on tenant RBStorageCluster. See Example 3-10.

Example 3-10 Creating a key for cluster RB_Storage_Cluster

```
[root@scale1 ~]# mmkeyserv key create --server gklm01.cloud.stg.forum.ibm.com
--tenant RBStorageCluster
Enter password for the key server gklm01.cloud.stg.forum.ibm.com:
KEY-29e4e69-f3bb22ca-eb9e-4e61-874c-8fbef532c12d
[root@scale1 ~]#
```

3.3.2 RB_Remote_Cluster.cloud.stg.forum.ibm.com cluster

Add the GKLM servers GKLM01 and GKLM02 to the encryption configuration on cluster RB_Remote_Cluster.

Perform the following commands from any of the nodes in the cluster.

1. Add key servers to IBM Storage Scale cluster RB_Remote_Cluster. See Example 3-11.

Example 3-11 Adding key servers to RB_Remote_Cluster

```
[root@RemoteScale1 ~]# mmkeyserv server add gklm01.cloud.stg.forum.ibm.com
--backup gklm02.cloud.stg.forum.ibm.com
Enter password for the key server gklm01.cloud.stg.forum.ibm.com:
The security certificate(s) from gklm01.cloud.stg.forum.ibm.com must be accepted
to continue. View the certificate(s) to determine whether you want to trust the
certifying authority.
Do you want to view or trust the certificate(s)? (view/yes/no) view

Serial number:          341fb4ab
SHA-256 digest:
6029ed54a62f10bd85671aa60f72b6bc91859229cea52481566814e59c62a8bd
Signature:
3bf06cc609e8d57de352a496346572096fb230d9a0b9c209154957f03484715f14157fe3104214f7c0
721f412c8f7ad093f32ed23ba8034b02473576da944d22799c1978fdae28729210faee97687b02ce6f
```

```
afe666b77d57b7f4e69eae816b40c9fa3f5af3650af63273ad5f4f827b80ac76a3cc1b089b9c3c8748
ba595f872a96208647a1e44467e9a256910abe50d0eb159461e7f580a44c74b6dc785ada7670e6f3cc
a91202090d3edd9253192f56c18f0e99e49c4e24976954f9112238e1cf37370ee4ecc1ef72045f8371
9cff3b07d094da16aa0be5ec9f13b86f834f76dad17e3564261e1e5e463dfcb4e4f757717574cb1e2c
a87ed0ef8e018986975b
```

```
Signature algorithm:  SHA256WithRSASignature
Key size:              2048
Issuer:               C=us, O=ibm, OU=gklm42server, CN=localhost
Subject:             C=us, O=ibm, OU=gklm42server, CN=localhost
Valid from:          Aug 04 12:28:52 2023 EEST (+0300)
Valid until:         Aug 03 12:28:52 2024 EEST (+0300)
```

```
Do you trust the certificate(s) above? (yes/no) yes
mmkeyserv: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
[root@RemoteScale1 ~]#
```

2. Verify keyserver configuration on RB_Remote_Cluster. See Example 3-12.

Example 3-12 Verifying keyserver configuration on RB_Remote_Cluster

```
root@RemoteScale1 ~]# mmkeyserv server show
gklm01.cloud.stg.forum.ibm.com
Type:                ISKLM
IPA:                 10.134.184.59
User ID:             SKLMAdmin
REST port:           9443
Label:               1_gklm01
NIST:                on
FIPS1402:            off
Backup Key Servers:  gklm02.cloud.stg.forum.ibm.com
Distribute:          yes
Retrieval Timeout:   60
Retrieval Retry:     3
Retrieval Interval:  10000
REST Certificate Expiration: 2024-08-03 12:28:52 (+0300)
KMIP Certificate Expiration:
```

```
[root@RemoteScale1 ~]#
```

3. Add a tenant (Device Group) RBRemoteCluster to server GKLM01. See Example 3-13.

Example 3-13 Adding a tenant (Device Group) RBRemoCluster to GKLM server GKLM01

```
[root@RemoteScale1 ~]# mmkeyserv tenant add RBRemoteCluster --server
gklm01.cloud.stg.forum.ibm.com
Enter password for the key server gklm01.cloud.stg.forum.ibm.com:
mmkeyserv: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
[root@RemoteScale1 ~]#
```

4. Register IBM Storage Scale cluster RB_Remote_Cluster as a client on server GKLM01 with name RBClientCluster2. See Example 3-14.

Example 3-14 Registering cluster RB_Remote_Cluster as a client on server GKLM01

```
[root@RemoteScale1 ~]# mmkeyserv client create RBClientCluster2 --server
gklm01.cloud.stg.forum.ibm.com
Enter password for the key server gklm01.cloud.stg.forum.ibm.com:
Create a pass phrase for keystore: <securely store this password. It is needed
later>
Confirm your pass phrase: <securely store this password. It is needed later>
mmkeyserv: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

5. Register client RBClientCluster2 to a tenant RBRemoteCluster. See Example 3-15.

Example 3-15 Registering client RBClientCluster2 to a tenant RBRemoteCluster

```
[root@RemoteScale1 ~]# mmkeyserv client register RBClientCluster2 --tenant
RBRemoteCluster --rkm-id NewYork_Tokyo
Enter password for the key server of client RBClientCluster2:
mmkeyserv: [I] Client currently does not have access to the key. Continue the
registration process ...
mmkeyserv: Successfully accepted client certificate
mmkeyserv: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
[root@RemoteScale1 ~]#
```

6. Create a key for cluster RB_Remote_Cluster on tenant RBRemoteCluster. See Example 3-16.

Example 3-16 Creating a key for cluster RB_Remote_Cluster on tenant RBRemoteCluster

```
[root@RemoteScale1 ~]# mmkeyserv key create --server
gklm01.cloud.stg.forum.ibm.com --tenant RBRemoteCluster
Enter password for the key server gklm01.cloud.stg.forum.ibm.com:
KEY-29e4e69-3fad8cca-5df8-401a-9271-ca049631ca6e
[root@RemoteScale1 ~]#
```

3.3.3 RB_Muumilaakso.cloud.stg.forum.ibm.com cluster

Add the GKLM servers GKLM01 and GKLM02 to encryption configuration on cluster RB_Muumilaakso.

Perform the following commands from any of the nodes in the cluster.

1. Add keyservers to IBM Storage Scale cluster RB_Muumilaakso. See Example 3-17.

Example 3-17 Adding keyservers to IBM Storage Scale cluster RB_Muumilaakso

```
[[root@moomi ~]# mmkeyserv server add gklm01.cloud.stg.forum.ibm.com --backup
gklm02.cloud.stg.forum.ibm.com
Enter password for the key server gklm01.cloud.stg.forum.ibm.com:
The security certificate(s) from gklm01.cloud.stg.forum.ibm.com must be accepted
to continue. View the certificate(s) to determine whether you want to trust the
certifying authority.
Do you want to view or trust the certificate(s)? (view/yes/no) view
```

```

Serial number:          341fb4ab
SHA-256 digest:
6029ed54a62f10bd85671aa60f72b6bc91859229cea52481566814e59c62a8bd
Signature:
3bf06cc609e8d57de352a496346572096fb230d9a0b9c209154957f03484715f14157fe3104214f7c0
721f412c8f7ad093f32ed23ba8034b02473576da944d22799c1978fdae28729210faee97687b02ce6f
afe666b77d57b7f4e69eae816b40c9fa3f5af3650af63273ad5f4f827b80ac76a3cc1b089b9c3c8748
ba595f872a96208647a1e44467e9a256910abe50d0eb159461e7f580a44c74b6dc785ada7670e6f3cc
a91202090d3edd9253192f56c18f0e99e49c4e24976954f9112238e1cf37370ee4ecc1ef72045f8371
9cff3b07d094da16aa0be5ec9f13b86f834f76dad17e3564261e1e5e463dfcb4e4f757717574cb1e2c
a87ed0ef8e018986975b
Signature algorithm:    SHA256WithRSASignature
Key size:               2048
Issuer:                 C=us, O=ibm, OU=gklm42server, CN=localhost
Subject:                C=us, O=ibm, OU=gklm42server, CN=localhost
Valid from:             Aug 04 12:28:52 2023 EEST (+0300)
Valid until:            Aug 03 12:28:52 2024 EEST (+0300)

```

Do you trust the certificate(s) above? (yes/no) yes

mmkeyserv: mmsdrfs propagation completed.

[root@moomi ~]#

2. Verify keyserver configuration on RB_Muumilaakso. See Example 3-18.

Example 3-18 Verifying keyserver configuration on RB_Muumilaakso

```

[root@moomi ~]# mmkeyserv server show
gklm01.cloud.stg.forum.ibm.com
      Type:                ISKLM
      IPA:                  10.134.184.59
      User ID:              SKLMAdmin
      REST port:            9443
      Label:                1_gklm01
      NIST:                  on
      FIPS1402:              off
      Backup Key Servers:   gklm02.cloud.stg.forum.ibm.com
      Distribute:            yes
      Retrieval Timeout:    60
      Retrieval Retry:      3
      Retrieval Interval:   10000
      REST Certificate Expiration: 2024-08-03 12:28:52 (+0300)
      KMIP Certificate Expiration:

```

[root@moomi ~]#

3. Add a Tenant (Device Group) Muumilaakso to GKLM server GKLM01. See Example 3-19.

Example 3-19 Adding a Tenant (Device Group) Muumilaakso to GKLM server GKLM01

```

[[root@moomi ~]# mmkeyserv tenant add RBMuumilaakso --server
gklm01.cloud.stg.forum.ibm.com
Enter password for the key server gklm01.cloud.stg.forum.ibm.com:
mmkeyserv: mmsdrfs propagation completed.
[root@moomi ~]#

```


4. Register IBM Storage Scale cluster RB_Muumilaakso as a client on server GKLM01 with name RBClientCluster3. See Example 3-20.

Example 3-20 Registering cluster RB_Muumilaakso as a client on server GKLM01

```
[root@moomi ~]# mmkeyserv tenant add RBMuumilaakso --server
gklm01.cloud.stg.forum.ibm.com
Enter password for the key server gklm01.cloud.stg.forum.ibm.com:
mmkeyserv: mmsdrfs propagation completed.
[root@moomi ~]#
[root@moomi ~]# mmkeyserv client create RBClientCluster3 --server
gklm01.cloud.stg.forum.ibm.com
Enter password for the key server gklm01.cloud.stg.forum.ibm.com:
Create a pass phrase for keystore: <securely store this password. It is needed
later>
Confirm your pass phrase: <securely store this password. It is needed later>
mmkeyserv: mmsdrfs propagation completed.
[root@moomi ~]#
```

5. Register the client RBClientCluster3 to a tenant RBMuumilaakso. See Example 3-21.

Example 3-21 Registering client RBClientCluster3 to a tenant RBMuumilaakso

```
[[root@moomi ~]# mmkeyserv client register RBClientCluster3 --tenant RBMuumilaakso
--rkm-id Muumitalo
Enter password for the key server of client RBClientCluster3:
mmkeyserv: [I] Client currently does not have access to the key. Continue the
registration process ...
mmkeyserv: Successfully accepted client certificate
mmkeyserv: mmsdrfs propagation completed.
[root@moomi ~]#
```

6. Create a key for cluster RB_Muumilaakso on tenant RBMuumilaakso. See Example 3-22.

Example 3-22 Creating a key for cluster RB_Muumilaakso on tenant RBMuumilaakso

```
[root@moomi ~]# mmkeyserv key create --server gklm01.cloud.stg.forum.ibm.com
--tenant RBMuumilaakso
Enter password for the key server gklm01.cloud.stg.forum.ibm.com:
KEY-29e4e69-2b459e17-a082-42ce-b619-93b7e64c0dad
[root@moomi ~]#
```

3.3.4 Confirming configuration on cluster RB_Storage_Cluster

Perform the following commands from any of the nodes in the IBM Storage Scale cluster.

1. Show the registered client settings for RB_Storage_Cluster. See Example 3-23.

Example 3-23 Showing the registered client settings for RB_Storage_Cluster

```
[root@scale1 ~]# mmkeyserv client show
RBClientCluster1
Label: RBClientCluster1
Key Server: gklm01.cloud.stg.forum.ibm.com
Tenants: RBStorageCluster
Certificate Expiration: 2026-08-22 16:09:25 (+0300)
Certificate Type: system-generated
```

```
[root@scale1 ~]#
```

2. Show the registered tenants for RB_Storage_Cluster. See Example 3-24.

Example 3-24 Showing the registered tenants

```
[root@scale1 ~]# mmkeyserv tenant show
RBStorageCluster
      Key Server:      gklm01.cloud.stg.forum.ibm.com
      Registered Client: RBClientCluster1
      RKM Id:          Turku_Jarvenpaa
[root@scale1 ~]#
```

3. Show the registered remote key manager (RKM) for RB_Storage_Cluster. See Example 3-25.

Example 3-25 Showing the registered RKM

```
[root@scale1 ~]# mmkeyserv rkm show
Turku_Jarvenpaa {
  type = ISKLM
  kmipServerUri = tls://gklm02.cloud.stg.forum.ibm.com:5696
  kmipServerUri2 = tls://10.134.184.59:5696
  keyStore = /var/mmfs/ssl/keyServ/serverK mip.1_gklm01.RBClientCluster1.1.p12
  passphrase = omelPOR2per3
  clientCertLabel = RBClientCluster1
  tenantName = RBStorageCluster
}
[root@scale1 ~]#
```

3.3.5 Confirming configuration on cluster RB_Remote_Cluster

Confirm configurations on the remote cluster.

1. Show the registered client settings for RB_Remote_Cluster cluster. See Example 3-26.

Example 3-26 Showing the registered client settings

```
[root@RemoteScale1 ~]# mmkeyserv client show
RBClientCluster2
      Label:          RBClientCluster2
      Key Server:      gklm01.cloud.stg.forum.ibm.com
      Tenants:         RBRemoteCluster
      Certificate Expiration: 2026-08-22 16:34:01 (+0300)
      Certificate Type: system-generated
[root@RemoteScale1 ~]#
```

2. Show the registered tenants for RB_Remote_Cluster. See Example 3-27.

Example 3-27 Showing the registered tenants for RB_Remote_Cluster

```
root@RemoteScale1 ~]# mmkeyserv tenant show
RBRemoteCluster
      Key Server:      gklm01.cloud.stg.forum.ibm.com
      Registered Client: RBClientCluster2
      RKM Id:          NewYork_Tokyo
```

```
[root@RemoteScale1 ~]#
```

3. Show the registered RKM for RB_Remote_Cluster. See Example 3-28

Example 3-28 Showing the registered RKM for RB_Remote_Cluster

```
root@RemoteScale1 ~]# mmkeyserv rkm show
NewYork_Tokyo {
  type = ISKLM
  kmipServerUri = tls://gklm02.cloud.stg.forum.ibm.com:5696
  kmipServerUri2 = tls://10.134.184.59:5696
  keyStore = /var/mmfs/ssl/keyServ/serverKmip.1_gklm01.RBClientCluster2.1.p12
  passphrase = kiss1K0IR2
  clientCertLabel = RBClientCluster2
  tenantName = RBRemoteCluster
}
[root@RemoteScale1 ~]#
```

3.3.6 Confirming configuration on IBM Storage Scale cluster RB_Muumilaakso.cloud.stg.forum.ibm.com

1. Show the registered client settings for RB_Muumilaakso cluster. See Example 3-29.

Example 3-29 Showing the registered client settings for RB_Muumilaakso cluster

```
[root@moomi ~]# mmkeyserv client show
RBClientCluster3
  Label:                RBClientCluster3
  Key Server:            gklm01.cloud.stg.forum.ibm.com
  Tenants:               RBMuumilaakso
  Certificate Expiration: 2026-08-23 15:39:50 (+0300)
  Certificate Type:      system-generated
[root@moomi ~]#
```

2. Show the registered tenants for RB_Muumilaakso cluster. See Example 3-30.

Example 3-30 Showing the registered tenants for RB_Muumilaakso cluster

```
[root@moomi ~]# mmkeyserv tenant show
RBMuumilaakso
  Key Server:            gklm01.cloud.stg.forum.ibm.com
  Registered Client:     RBClientCluster3
  RKM Id:                Muumitalo
[root@moomi ~]#
```

3. Show the registered RKM for cluster. See Example 3-31.

Example 3-31 Showing the registered RKM for RB_Muumilaakso cluster

```
[root@moomi ~]# mmkeyserv rkm show
Muumitalo {
  type = ISKLM
  kmipServerUri = tls://gklm02.cloud.stg.forum.ibm.com:5696
  kmipServerUri2 = tls://10.134.184.59:5696
  keyStore = /var/mmfs/ssl/keyServ/serverKmip.1_gklm01.RBClientCluster3.1.p12
  passphrase = pikku2myy10
}
```

```
clientCertLabel = RBClientCluster3
tenantName = RBMuumilaakso
}
[root@moomi ~]#
```

3.4 Configuring Remote clusters and remote cluster mounts

This section demonstrates how to configure remote clusters on IBM Storage Scale and mount file systems and filesets.

Always review current IBM documentation for remote mounts before configuring a new environment. For the latest version at the time of writing, see [Mounting a remote GPFS file system](#).

It is also possible to mount filesets instead of file systems on the remote clusters. For more information, see [mmauth command](#).

On the owningCluster system, the system administrator issues the **mmauth genkey** command to generate a public and private key pair. The key pair is placed in `/var/mmfs/ssl`. The public key file is `id_rsa.pub`.

1. Run the **mmauth genkey** command. See Example 3-32

Example 3-32 mmauth genkey command

```
[root@scale1 ~]# mmauth genkey new
mmauth: Command successfully completed
[root@scale1 ~]#
```

2. On the owningCluster system, the system administrator enables authorization by entering the **mmauth** command. See Example 3-33.

Example 3-33 mmauth update command

```
[root@scale1 ~]# mmauth update . -l AUTHONLY
mmauth: Propagating the cluster configuration data to all affected nodes.
mmauth: Command successfully completed
[root@scale1 ~]#
```

3. The system administrator of the owningCluster gives the file `/var/mmfs/ssl/id_rsa.pub` to the system administrator of the accessingCluster system. This operation requires the two administrators to coordinate their activities and must occur outside of the GPFS command environment.
4. The system administrator of accessingCluster can rename the key file and put it in any directory on the node on which the administrator is working. If the administrator provides the correct path and file name in the **mmremoteccluster add** command, the file can be used successfully. See Example 3-40 on page 53 to view the **mmremoteccluster add** command. In Example 3-34, the file `id_rsa_RB_Storage_Cluster.pub` is renamed to reflect the name of the cluster owning this key.

Example 3-34 transferring the id_rsa.pub key file between systems

```
[root@scale1 ~]# scp /var/mmfs/ssl/id_rsa_new.pub
root@RemoteScale1:/var/mmfs/ssl/id_rsa_RB_Storage_Cluster.pub
root@remotescale1's password:
```

```
id_rsa_new.pub
100% 1898    1.4MB/s   00:00
[root@scale1 ~]#

[root@scale1 ~]# scp /var/mmfs/ssl/id_rsa_new.pub
root@moomi:/var/mmfs/ssl/id_rsa_RB_Storage_Cluster.pub
root@moomi's password:
id_rsa_new.pub
100% 1898    674.3KB/s   00:00
[root@scale1 ~]#
```

5. On the accessingCluster system, the system administrator issues the **mmauth genkey** command to generate a public/private key pair. The key pair is placed in /var/mmfs/ssl. The public key file is id_rsa.pub. See Example 3-35.

Example 3-35 mmauth genkey command

```
[root@RemoteScale1 ~]# mmauth genkey new
mmauth: Command successfully completed
[root@RemoteScale1 ~]#

[root@moomi ~]# mmauth genkey new
mmauth: Command successfully completed
[root@moomi ~]#
```

6. On the accessingCluster system, the system administrator enables authorization by entering the following **mmauth update** command. See Example 3-36.

Example 3-36 mmauth update command

```
[root@RemoteScale1 ~]# mmauth update . -l AUTHONLY
mmauth: Propagating the cluster configuration data to all affected nodes.
mmauth: Command successfully completed
[root@RemoteScale1 ~]#

[root@moomi ~]# mmauth update . -l AUTHONLY
mmauth: Command successfully completed
[root@moomi ~]#
```

7. The system administrator of accessingCluster gives the key file /var/mmfs/ssl/id_rsa.pub to the system administrator of owningCluster. This operation requires the two administrators to coordinate their activities and occurs outside of the GPFS command environment. The files can be renamed. See Example 3-37.

Example 3-37 transfer the id_rsa.pub key file between systems

```
[root@RemoteScale1 ~]# scp /var/mmfs/ssl/id_rsa_new.pub
root@Scale1:/var/mmfs/ssl/id_rsa_RB_Remote_Cluster.pub
root@scale1's password:
id_rsa_new.pub
100% 1891    950.4KB/s   00:00
[root@RemoteScale1 ~]#

[root@moomi ~]# scp /var/mmfs/ssl/id_rsa_new.pub
root@Scale1:/var/mmfs/ssl/id_rsa_RB_Muumilaakso.pub
root@scale1's password:
```

```
id_rsa_new.pub
100% 1881    1.0MB/s   00:00
[root@moomi ~]#
```

The system administrator of the owningCluster can rename the key file and put it in any available directory on the node on which they are working. In Example 3-37 on page 51, the key files are renamed to `id_rsa_RB_Remote_Cluster.pub` and `id_rsa_RB_Muumilaakso.pub` during the `scp` transfer to reflect the name of the cluster that owns this key.

On the owningCluster system run the `mmauth add` command to authorize accessingCluster to mount file systems that are owned by owningCluster. This is done by using the key file that was received from the administrator of the accessingCluster system. See Example 3-38.

Example 3-38 mmauth add command to enable cross-node access

```
[root@scale1 ~]# mmauth add RB_Remote_Cluster.cloud.stg.forum.ibm.com -k
/var/mmfs/ssl/id_rsa_RB_Remote_Cluster.pub
mmauth: Propagating the cluster configuration data to all affected nodes.
mmauth: [I] The tscCmdAllowRemoteConnections configuration parameter on the local
cluster has value "no". If the remote cluster has cluster release level
(minReleaseLevel) less than 5.1.3.0, change the value of
tscCmdAllowRemoteConnections in the local cluster to "yes".
mmauth: Command successfully completed
[root@scale1 ~]#
[root@scale1 ~]# mmauth add RB_Muumilaakso.cloud.stg.forum.ibm.com -k
/var/mmfs/ssl/id_rsa_RB_Muumilaakso.pub
mmauth: Propagating the cluster configuration data to all affected nodes.
mmauth: [I] The tscCmdAllowRemoteConnections configuration parameter on the local
cluster has value "no". If the remote cluster has cluster release level
(minReleaseLevel) less than 5.1.3.0, change the value of
tscCmdAllowRemoteConnections in the local cluster to "yes".
mmauth: Command successfully completed
[root@scale1 ~]#
```

8. On the owningCluster system, run the `mmauth grant` command to authorize accessingCluster to mount specific file systems that are owned by the owningCluster system. See Example 3-39.

Example 3-39 mmauth grant command

```
[root@scale1 ~]# mmauth grant RB_Remote_Cluster.cloud.stg.forum.ibm.com -f fs01

mmauth: Granting cluster RB_Remote_Cluster.cloud.stg.forum.ibm.com access to file
system fs01:
        access type rw; root credentials will not be remapped.

mmauth: Propagating the cluster configuration data to all affected nodes.
mmauth: Command successfully completed
[root@scale1 ~]#
[root@scale1 ~]# mmauth grant RB_Muumilaakso.cloud.stg.forum.ibm.com -f fs01

mmauth: Granting cluster RB_Muumilaakso.cloud.stg.forum.ibm.com access to file
system fs01:
        access type rw; root credentials will not be remapped.

mmauth: Propagating the cluster configuration data to all affected nodes.
```

```
mmauth: Command successfully completed
[root@scale1 ~]#
```

Note: When you use IBM Storage Scale, instead of mounting a complete file system, you can mount filesets.

If the accessing cluster is mounting the remote file system in read-only mode, then only a subset of the events is generated. For more information, see [File audit logging events](#).

9. On `accessingCluster`, use the `mmremoteccluster add` command to define the cluster name, contact nodes and public key for `owningCluster`. See Example 3-40.

Example 3-40 mmremoteccluster add command

```
[root@RemoteScale1 ~]# mmremoteccluster add
RB_Storage_Cluster.cloud.stg.forum.ibm.com -n Scale1,Scale2,Scale3 -k
/var/mmfs/ssl/id_rsa_RB_Storage_Cluster.pub
mmremoteccluster: [I] The tscCmdAllowRemoteConnections configuration parameter on
the local cluster has value "no". If the remote cluster has cluster release level
(minReleaseLevel) less than 5.1.3.0, change the value of
tscCmdAllowRemoteConnections in the local cluster to "yes".
mmremoteccluster: Command successfully completed
mmremoteccluster: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
[root@RemoteScale1 ~]#
```

```
[root@moomi ~]# mmremoteccluster add RB_Storage_Cluster.cloud.stg.forum.ibm.com -n
Scale1,Scale2,Scale3 -k /var/mmfs/ssl/id_rsa_RB_Storage_Cluster.pub
mmremoteccluster: [I] The tscCmdAllowRemoteConnections configuration parameter on
the local cluster has value "no". If the remote cluster has cluster release level
(minReleaseLevel) less than 5.1.3.0, change the value of
tscCmdAllowRemoteConnections in the local cluster to "yes".
mmremoteccluster: Command successfully completed
mmremoteccluster: mmsdrfs propagation completed.
[root@moomi ~]#
```

10. By using the `mmremotefs` command, the system administrator of `accessingCluster` can locate the serving cluster and mount its file systems.

On `accessingCluster`, `RemoteScale1` and `moomi`, run one or more `mmremotefs` commands to identify the file systems in `owningCluster` that are to be accessed by nodes in `accessingCluster`. See Example 3-41.

Example 3-41 mmremotefs add command

```
[root@RemoteScale1 ~]# mmremotefs add fs01 -f fs01 -C
RB_Storage_Cluster.cloud.stg.forum.ibm.com -T /ibm/fs01
mmremotefs: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
[root@RemoteScale1 ~]#
```

```
[root@moomi ~]# mmremotefs add fs01 -f fs01 -C
RB_Storage_Cluster.cloud.stg.forum.ibm.com -T /ibm/fs01
mmremotefs: mmsdrfs propagation completed.
[root@moomi ~]#
```

11. Mount the file systems on the remote clusters using the **mmount** command. See Example 3-42.

Example 3-42 mmount command

```
[root@RemoteScale1 ~]# mmmount fs01
Wed Sep  6 10:54:37 EEST 2023: mmmount: Mounting file systems ...
[root@RemoteScale1 ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
devtmpfs	16G	0	16G	0%	/dev
tmpfs	16G	0	16G	0%	/dev/shm
tmpfs	16G	17M	16G	1%	/run
tmpfs	16G	0	16G	0%	/sys/fs/cgroup
/dev/mapper/rhel_remotescale1-root	48G	6.5G	41G	14%	/
/dev/mapper/rhel_remotescale1-home	24G	3.2G	20G	14%	/home
/dev/sda2	1014M	226M	789M	23%	/boot
/dev/sda1	599M	5.8M	594M	1%	/boot/efi
tmpfs	3.2G	0	3.2G	0%	/run/user/0
fs01	300G	45G	256G	15%	/ibm/fs01

```
[root@RemoteScale1 ~]#

[root@moomi ~]# mmmount fs01
Wed Sep  6 10:55:47 EEST 2023: mmmount: Mounting file systems ...
[root@moomi ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
devtmpfs	7.9G	0	7.9G	0%	/dev
tmpfs	7.9G	0	7.9G	0%	/dev/shm
tmpfs	7.9G	17M	7.9G	1%	/run
tmpfs	7.9G	0	7.9G	0%	/sys/fs/cgroup
/dev/mapper/rhel_moomi-root	70G	18G	53G	25%	/
/dev/mapper/rhel_moomi-home	49G	379M	49G	1%	/home
/dev/sda2	1014M	228M	787M	23%	/boot
/dev/sda1	599M	5.8M	594M	1%	/boot/efi
tmpfs	1.6G	0	1.6G	0%	/run/user/0
fs01	300G	45G	256G	15%	/ibm/fs01

```
[root@moomi ~]#
```

3.5 Enabling encryption on selected file systems and filesets

Encryption in IBM Storage Scale is enabled by defining a set of rules in a policy file. The policy file is installed on the file system that contains the filesets and data to be encrypted.

A file system can have only one active policy installed. Adding encryption rules into an existing file system can be done by adding the encryption rules to the existing policy file and installing the policy for the file system with the **mmchpolicy** command. The **mmchpolicy** command is shown in Example 3-50 on page 58.

To review or confirm the file system current policy name, installation time and “installed by” information use the **mmfspolicy filesystem** command. See Example 3-43.

Example 3-43 mmfspolicy command

```
[root@scale1 ~]# mmfspolicy fs01
No policy file was installed for file system 'fs01'.
Data will be stored in pool 'system'.
```

```
[root@scale1 ~]#
```

Encryption on IBM Storage Scale can be implemented either on a file system or a fileset level. For this environment, the encryption of data is defined to the fileset with an encryption policy enabled in a parent file system.

When no encryption policy is applied to a file system or to a fileset within a file system, the file system attribute **encryption** is set to its default value of **no**. This can be checked using the **mmfsfs** command. See Example 3-44.

Example 3-44 mmfsfs command output

[root@scale1 ~]# mmfsfs fs01 --encryption		
flag	value	description

--encryption	no	Encryption enabled?
[root@scale1 ~]#		

When encryption is enabled for the first time either on a file system or on a member file set, the attribute for encryption for the entire file system is set to **enabled**. This setting cannot be reverted after it is enabled.

Note: After the encryption attribute for the file system has been set to **enabled**, any client node mounting the file system must be running a version of Storage Scale that supports encryption.

If the version of IBM Storage Scale that is installed on the client nodes does not support the encryption feature, the nodes cannot mount the entire file system whether the encryption policy includes that file system.

GKLM infrastructure is critical to prevent the loss of access or loss of data. If you lose access to the GKLM server completely, there is no way to recover your data from the IBM Storage Scale file system. If the key server is lost, data can be recovered from only a restored data backup.

By default, the GKLM server is contacted at the time that the file system is mounted and then every time the *encryptionKeyCacheExpiration* seconds pass. The default is 900 seconds, which is 15 minutes. If a node cannot reach the GKLM server, it loses access to the encrypted file system. When setting *encryptionKeyCacheExpiration* to 0, the check with SKLM happens only during the file system mount. Certain security implications apply if *encryptionKeyCacheExpiration* is set to 0 and there is no rechecking of access to the key server.

It is suggested that when using GKLM, follow the “four eyes” principle. This means that no single person or team has access to both the GKLM system and the IBM Storage Scale nodes.

3.5.1 Creating encryption policy for a fileset

An IBM Storage Scale policy rule is a clear text file with all policies listed in the preferred order. The IBM Storage Scale policy engine applies the policy file rules on a file system level. Different file systems can have different policies associated with them.

1. Edit a file to create a policy file for the file system.

Example 3-45 Open a policy file for the file system

```
[root@scale1 GPFSPolicies]# vi /root/GPFSPolicies/FinlandEnc.policy
[root@scale1 GPFSPolicies]#
```

Example 3-46 list the contents of the policy file FinlandEnc.policy. Installation of the policy is shown in Example 3-50 on page 58.

Example 3-46 Add rules to the policy file

```
RULE 'ENC_Turku' SET ENCRYPTION 'Turku_ENC' FOR FILESET('Turku')
RULE 'ENC_Turku_KEYS' ENCRYPTION 'Turku_ENC' IS ALGO 'DEFAULTNISTSP800131A'
KEYS('KEY-29e4e69-f3bb22ca-eb9e-4e61-874c-8fbeb532c12d:Turku_Jarvenpaa')
RULE 'ENC_Järvenpää' SET ENCRYPTION 'Järvenpää_ENC' FOR FILESET('Järvenpää')
RULE 'ENC_Järvenpää_KEYS' ENCRYPTION 'Järvenpää_ENC' IS ALGO
'DEFAULTNISTSP800131A'
KEYS('KEY-29e4e69-f3bb22ca-eb9e-4e61-874c-8fbeb532c12d:Turku_Jarvenpaa')
```

Note: Use the correct RKM ID for the key at the end of the key string, separated with “:”-sign. RKM ID is displayed during the key creation in Example 3-10 on page 43.

3.5.2 Checking the installed policy for a file system

To check which policy is installed on the file system and its encryption status, use the **mmlspolicy -L** command. To list the complete details of the policy installed for a file system, run the command **mmlspolicy filesystem -L**. See Example 3-47.

Example 3-47 mmlspolicy -L output for the filespace

```
[root@scale1 fs01]# mmlspolicy fs01 -L
No policy file was installed for file system 'fs01'.
Data will be stored in pool 'system'.
[root@scale1 fs01]# mmlsattr fs01

[root@scale1 fs01]# mmlsfs fs01 --encryption
flag                value                description
-----
--encryption        no                Encryption enabled?
[root@scale1 fs01]#
```

To check encryption status of files stored on filesets Turku, Järvenpää, New_York and Tokyo on file system fs01 use the **mmlsattr -L** command:

Example 3-48 mmlsattr -L command output

```
[root@scale1 ~]# mmlsattr -L
/ibm/fs01/Järvenpää/Spectrum_Scale_Data_Management-5.1.7.1-x86_64-Linux-install |
grep -i encrypted
Encrypted:          no
[root@scale1 ~]# mmlsattr -L /ibm/fs01/Turku/Nuuska_Muikkusen_osoite_etelässä.txt
| grep -i encrypted
Encrypted:          no
[root@scale1 ~]# mmlsattr -L /ibm/fs01/Turku/Muumimamman_Pipariresepri.txt | grep
-i encrypted
Encrypted:          no
```

```
[root@scale1 ~]# mmlsattr -L
/ibm/fs01/New_York/RHEL-9.2.0-20230414.17-ppc64le-dvd1.iso | grep -i encrypted
Encrypted:          no
[root@scale1 ~]# mmlsattr -L /ibm/fs01/Tokyo/SLE-15-SP5-Full-ppc64le.iso | grep -i
encrypted
Encrypted:          no
[root@scale1 ~]# mmlsattr -L /ibm/fs01/Muumitalo/Muumimamman_Pipariresepti.txt |
grep -i encrypted
Encrypted:          no
[root@scale1 ~]#
```

3.5.3 Testing and changing the policy of a file system

Example 3-49 shows how to test and install a previously created encryption policy on file system fs01 and filesets Turku and Järvenpää.

Example 3-49 mmapplypolicy testing

```
[root@scale1 ~]# mmapplypolicy fs01 -I test -P
/root/GPFSPolicies/FinlandEnc.policy
[I] GPFS Current Data Pool Utilization in KB and %
Pool_Name          KB_Occupied      KB_Total  Percent_Occupied
system             46886912      314572800  14.904947917%
[I] 4063 of 915456 inodes used: 0.443823%.
[I] Loaded policy rules from /root/GPFSPolicies/FinlandEnc.policy.
Evaluating policy rules with CURRENT_TIMESTAMP = 2023-09-06@09:29:28 UTC
Parsed 4 policy rules.
[W] Attention: It seems there are no effective nor useful rules. You may want to
terminate this command!
RULE 'ENC_Turku' SET ENCRYPTION 'Turku_ENC' FOR FILESET('Turku')
RULE 'ENC_Turku_KEYS' ENCRYPTION 'Turku_ENC' IS ALGO 'DEFAULTNISTSP800131A'
KEYS('KEY-29e4e69-f3bb22ca-eb9e-4e61-874c-8fbeb532c12d:Turku_Jarvenpaa')
RULE 'ENC_Järvenpää' SET ENCRYPTION 'Järvenpää_ENC' FOR FILESET('Järvenpää')
RULE 'ENC_Järvenpää_KEYS' ENCRYPTION 'Järvenpää_ENC' IS ALGO
'DEFAULTNISTSP800131A'
KEYS('KEY-29e4e69-f3bb22ca-eb9e-4e61-874c-8fbeb532c12d:Turku_Jarvenpaa')

[I] 2023-09-06@09:29:28.507 Directory entries scanned: 33.
[I] Directories scan: 24 files, 9 directories, 0 other objects, 0 'skipped' files
and/or errors.
[I] 2023-09-06@09:29:31.948 Parallel-piped sort and policy evaluation. 33 files
scanned.
[I] 2023-09-06@09:29:31.968 Piped sorting and candidate file choosing. 0 records
scanned.
[I] Summary of Rule Applicability and File Choices:
  Rule#      Hit_Cnt      KB_Hit      Chosen      KB_Chosen
KB_I11      Rule

[I] Filesystem objects with no applicable rules: 22.

Predicted Data Pool Utilization in KB and %:
Pool_Name          KB_Occupied      KB_Total  Percent_Occupied
system             46886912      314572800  14.904947917%
[root@scale1 ~]#
```

Note: If any errors or unwanted actions are listed during the test of the policy, correct the errors in the policy file and recheck for any possible unwanted actions before assigning a policy to the file system.

Activate and confirm the policy on file system `fs01` as shown in Example 3-50.

Example 3-50 mmchpolicy command

```
[root@scale1 ~]# mmchpolicy fs01 /root/GPFSPolicies/FinlandEnc.policy
Validated policy 'FinlandEnc.policy': Parsed 4 policy rules.
Policy 'FinlandEnc.policy' installed and broadcast to all nodes.

[root@scale1 ~]# mmlspolicy fs01 -L
RULE 'ENC_Turku' SET ENCRYPTION 'Turku_ENC' FOR FILESET('Turku')
RULE 'ENC_Turku_KEYS' ENCRYPTION 'Turku_ENC' IS ALGO 'DEFAULTNISTSP800131A'
KEYS('KEY-29e4e69-f3bb22ca-eb9e-4e61-874c-8fbeb532c12d:Turku_Jarvenpaa')
RULE 'ENC_Jarvenpää' SET ENCRYPTION 'Järvenpää_ENC' FOR FILESET('Järvenpää')
RULE 'ENC_Jarvenpää_KEYS' ENCRYPTION 'Järvenpää_ENC' IS ALGO
'DEFAULTNISTSP800131A'
KEYS('KEY-29e4e69-f3bb22ca-eb9e-4e61-874c-8fbeb532c12d:Turku_Jarvenpaa')

[root@scale1 ~]#
```

3.5.4 Verifying and testing the encryption policy

Example 3-51 shows how to check the encryption status of file system `fs01` and existing files on a file system.

Example 3-51 Use mmlsfs and mmlsattr commands to check encryption status

```
[root@scale1 ~]# mmlsfs fs01 --encryption
flag          value          description
-----
--encryption  yes          Encryption enabled?
[root@scale1 ~]#

[root@scale1 ~]# mmlsattr -L
/ibm/fs01/Järvenpää/Spectrum_Scale_Data_Management-5.1.7.1-x86_64-Linux-install |
grep -i encrypted
Encrypted:          no
[root@scale1 ~]# mmlsattr -L /ibm/fs01/Turku/Nuuska_Muikkusen_osoite_etelässä.txt
| grep -i encrypted
Encrypted:          no
[root@scale1 ~]# mmlsattr -L /ibm/fs01/Turku/Muumimamman_Pipariresepri.txt | grep
-i encrypted
Encrypted:          no
[root@scale1 ~]# mmlsattr -L
/ibm/fs01/New_York/RHEL-9.2.0-20230414.17-ppc64le-dvd1.iso | grep -i encrypted
Encrypted:          no
[root@scale1 ~]# mmlsattr -L /ibm/fs01/Tokyo/SLE-15-SP5-Full-ppc64le.iso | grep -i
encrypted
Encrypted:          no
[root@scale1 ~]# mmlsattr -L /ibm/fs01/Muumitalo/Muumimamman_Pipariresepti.txt |
grep -i encrypted
```

```
Encrypted:          no
[root@scale1 ~]#
```

Note: After an encryption policy is applied to the file system, the file system attribute **encrypted** changes to yes.

In Example 3-51 on page 58, the attribute **encrypted** for files, which is already present for the file system, does not change when a change is made to the encryption policy of the file system.

Any new file that is created or any existing file that is modified after the policy for the file system is assigned or updated has its attribute **encrypted** set to yes.

Example 3-52 New and updated versus existing file encryption settings

```
[root@scale1 ~]# vi /ibm/fs01/Järvenpää/LB_address
[root@scale1 ~]# mmlsattr -L /ibm/fs01/Järvenpää/LB_address | grep -i Encrypted
Encrypted:          yes
[root@scale1 ~]# vi /ibm/fs01/Turku/LB_address
[root@scale1 ~]# mmlsattr -L /ibm/fs01/Turku/LB_address | grep -i Encrypted
Encrypted:          no
[root@scale1 ~]#
```

As shown in Example 3-52, after changing a file system policy to encrypt contents on filesets Turku and Järvenpää the files created on these filesets have their attribute **encrypted** set to yes.

Only IBM Storage Scale cluster nodes that have encryption keys defined for tenant RBStorageCluster are allowed to read or write files on filesets Turku and Järvenpää as shown in Example 3-53.

Example 3-53 File encryption settings and file access

```
root@RemoteScale1 ~]# touch /ibm/fs01/Järvenpää/NewFile
touch: setting times of '/ibm/fs01/Järvenpää/NewFile': No such file or directory

[root@RemoteScale1 ~]# touch /ibm/fs01/Turku/NewFile
[root@RemoteScale1 ~]# mmlsattr -L /ibm/fs01/Turku/NewFile | grep -i encrypted
Encrypted:          no
[root@RemoteScale1 ~]#

[root@RemoteScale1 ~]# cat /ibm/fs01/Järvenpää/LB_address
cat: /ibm/fs01/Järvenpää/LB_address: Operation not permitted

[root@RemoteScale1 ~]# cat /ibm/fs01/Turku/LB_address
LoadBalancer address:
10.134.184.62

[root@moomi Tokyo]# cd
[root@moomi ~]# cat /ibm/fs01/Järvenpää/LB_address
cat: /ibm/fs01/Järvenpää/LB_address: Operation not permitted

[root@moomi ~]# cat /ibm/fs01/Turku/LB_address
LoadBalancer address:
10.134.184.62
```

3.5.5 Changing a policy to include encryption for filesets

This section shows details and information about making changes to the policy to now include a configuration as shown in Figure 3-1.

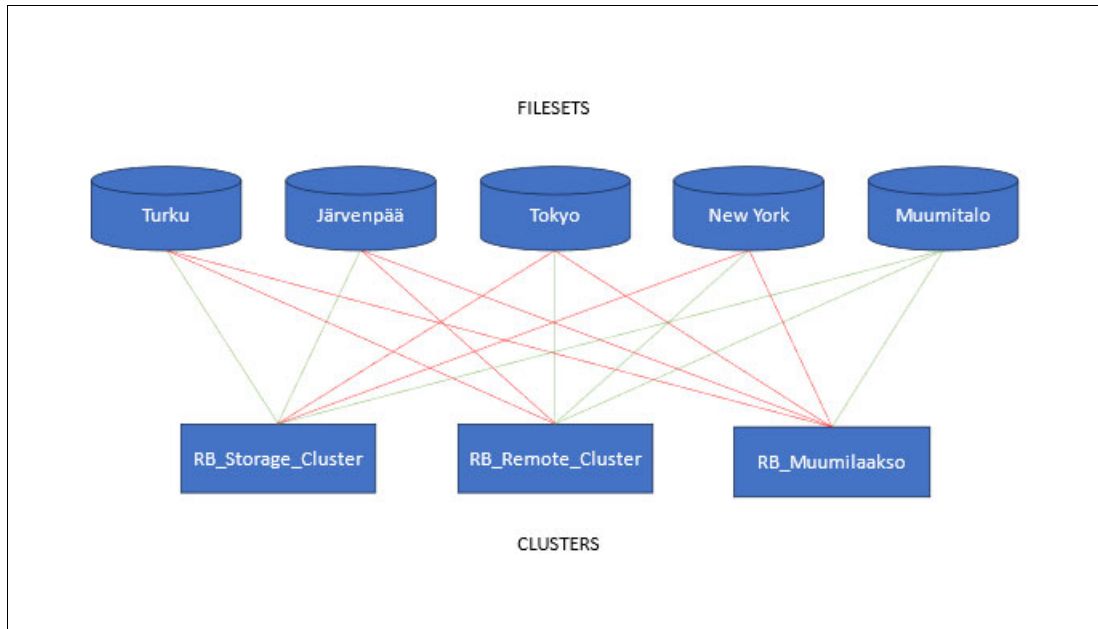


Figure 3-1 Cluster and filesets encryption relationship

After the commands that are described in this section are run, the configuration of the clusters matches Figure 3-1. The green lines between each cluster and fileset indicate that the cluster has access to this fileset. If a cluster is linked to a fileset with a red line, it cannot access the contents of those filesets.

To add encryption to filesets Tokyo and New_York, modify existing encryption policy FinlandEnc.policy to include encryption settings for filesets Tokyo and New_York with an encryption key for the IBM Storage Scale cluster RB_Remote_Cluster assigned to tenant RBRemoteCluster.

Optionally, you can create a new policy file, copy the content of the currently active policy into that file and then add additional rules for Tokyo and New_York encryption. If this is done, then the new policy file must be assigned to the file system so that the new policy is used instead of the current active policy.

Add the lines into an existing policy file. The contents of the modified policy file is shown in Example 3-54.

Example 3-54 Modified policy file contents

```
[root@scale1 ~]# cat /root/GPFSPolicies/FinlandEnc.policy
RULE 'ENC_Turku' SET ENCRYPTION 'Turku_ENC' FOR FILESET('Turku')
RULE 'ENC_Turku_KEYS' ENCRYPTION 'Turku_ENC' IS ALGO 'DEFAULTNISTSP800131A'
KEYS('KEY-29e4e69-f3bb22ca-eb9e-4e61-874c-8fbeb532c12d:Turku_Jarvenpaa')
RULE 'ENC_Järvenpää' SET ENCRYPTION 'Järvenpää_ENC' FOR FILESET('Järvenpää')
```

```

RULE 'ENC_Järvenpää_KEYS' ENCRYPTION 'Järvenpää_ENC' IS ALGO
'DEFAULTNISTSP800131A'
KEYS('KEY-29e4e69-f3bb22ca-eb9e-4e61-874c-8fbeb532c12d:Turku_Jarvenpaa')
RULE 'ENC_Tokyo' SET ENCRYPTION 'Tokyo_ENC' FOR FILESET('Tokyo')
RULE 'ENC_Tokyo_KEYS' ENCRYPTION 'Tokyo_ENC' IS ALGO 'DEFAULTNISTSP800131A'
KEYS('KEY-29e4e69-3fad8cca-5df8-401a-9271-ca049631ca6e:NewYork_Tokyo')
RULE 'ENC_York' SET ENCRYPTION 'New_York_ENC' FOR FILESET('New_York')
RULE 'ENC_York_KEYS' ENCRYPTION 'New_York_ENC' IS ALGO 'DEFAULTNISTSP800131A'
KEYS('KEY-29e4e69-3fad8cca-5df8-401a-9271-ca049631ca6e:NewYork_Tokyo')

[root@scale1 ~]#

```

Note: Encryption key for filesets Turku and Järvenpää is using a different encryption key than filesets Tokyo and New_York.

Before you apply the new policy to file system fs01, tenant RBRemoteCluster must be added to the configuration of cluster RBStorageCluster, which is the owner of the file system. See step 3 on page 44. Also, add the owning cluster RBStorageCluster as a client to the newly introduced tenant.

The process of adding the tenant and adding a client to the tenant is shown in Example 3-55.

Example 3-55 Adding a tenant to the cluster to allow file access

```

[root@scale1 ~]# mmkeyserv tenant add RBRemoteCluster --server
gklm01.cloud.stg.forum.ibm.com
Enter password for the key server gklm01.cloud.stg.forum.ibm.com:
mmkeyserv: [I] Tenant RBRemoteCluster already exists. Processing continues ...
mmkeyserv: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.

```

```

[root@scale1 ~]# mmkeyserv tenant show
RBStorageCluster
    Key Server:          gklm01.cloud.stg.forum.ibm.com
    Registered Client:   RBClientCluster1
    RKM Id:              Turku_Jarvenpaa
RBRemoteCluster
    Key Server:          gklm01.cloud.stg.forum.ibm.com
    Registered Client:   (none)
    RKM Id:              (none)

```

```

[root@scale1 ~]# mmkeyserv client register RBClientCluster1 --tenant
RBRemoteCluster --rkm-id NewYork_Tokyo
Enter password for the key server of client RBClientCluster1:
mmkeyserv: [I] Client currently does not have access to the key. Continue the
registration process ...
mmkeyserv: Successfully accepted client certificate
mmkeyserv: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
[root@scale1 ~]# mmkeyserv tenant show
RBStorageCluster
    Key Server:          gklm01.cloud.stg.forum.ibm.com
    Registered Client:   RBClientCluster1
    RKM Id:              Turku_Jarvenpaa
RBRemoteCluster

```

```

Key Server:          gklm01.cloud.stg.forum.ibm.com
Registered Client:   RBClientCluster1
RKM Id:              NewYork_Tokyo
[root@scale1 ~]#

```

Use the **mmchpolicy** command to apply a new policy to a file system fs01.

Example 3-56 Applying a new policy to a file system

```

[root@scale1 ~]# mmchpolicy fs01 /root/GPFSPolicies/FinlandEnc.policy -I yes
Validated policy 'FinlandEnc.policy': Parsed 8 policy rules.
Policy `FinlandEnc.policy' installed and broadcast to all nodes.
[root@scale1 ~]# mmlspolicy fs01 -L
RULE 'ENC_Turku' SET ENCRYPTION 'Turku_ENC' FOR FILESET('Turku')
RULE 'ENC_Turku_KEYS' ENCRYPTION 'Turku_ENC' IS ALGO 'DEFAULTNISTSP800131A'
KEYS('KEY-29e4e69-f3bb22ca-eb9e-4e61-874c-8fbeb532c12d:Turku_Jarvenpaa')
RULE 'ENC_Järvenpää' SET ENCRYPTION 'Järvenpää_ENC' FOR FILESET('Järvenpää')
RULE 'ENC_Järvenpää_KEYS' ENCRYPTION 'Järvenpää_ENC' IS ALGO
'DEFAULTNISTSP800131A'
KEYS('KEY-29e4e69-f3bb22ca-eb9e-4e61-874c-8fbeb532c12d:Turku_Jarvenpaa')
RULE 'ENC_Tokyo' SET ENCRYPTION 'Tokyo_ENC' FOR FILESET('Tokyo')
RULE 'ENC_Tokyo_KEYS' ENCRYPTION 'Tokyo_ENC' IS ALGO 'DEFAULTNISTSP800131A'
KEYS('KEY-29e4e69-3fad8cca-5df8-401a-9271-ca049631ca6e:NewYork_Tokyo')
RULE 'ENC_York' SET ENCRYPTION 'New_York_ENC' FOR FILESET('New_York')
RULE 'ENC_York_KEYS' ENCRYPTION 'New_York_ENC' IS ALGO 'DEFAULTNISTSP800131A'
KEYS('KEY-29e4e69-3fad8cca-5df8-401a-9271-ca049631ca6e:NewYork_Tokyo')
[root@scale1 ~]#

```

After the new policy has been applied to file system fs01 all new files created on filesets Turku, Järvenpää, Tokyo and New_York are encrypted. Files created on other filesets on file system fs01 are not encrypted.

Access to the encrypted filesets is allowed from all nodes of clusters RBStorageCluster and RBRemoteCluster.

A 3rd cluster that remotely mounts file system fs01 from RBStorageCluster does not have access to any encrypted content.

3.5.6 File system and fileset access summary

After all configuration steps are completed, the clusters can access the contents of file system fs01 and filesets as shown in Table 3-3 and as illustrated in Figure 3-1 on page 60.

Table 3-3 Cluster and fileset access

Fileset	IBM Storage Scale Cluster		
	RB_Storage_Cluster	RB_Remote_Cluster	RB_Muumilaakso
Turku	Yes	No	No
Järvenpää	Yes	No	No
Tokyo	Yes	Yes	No

	IBM Storage Scale Cluster		
Fileset	RB_Storage_Cluster	RB_Remote_Cluster	RB_Muumilaakso
New_York	Yes	Yes	No
Muumitalo	Yes	Yes	Yes

3.6 File system and file access and attributes of encrypted files

When encryption is enabled for the IBM Storage Scale environment by installing a policy for a file system, the following changes occur:

- Any node that mounts an encrypted file system or a file system that contains a fileset that is enabled for encryption must run an version of IBM Storage Scale that supports encryption. A node without the `gpfs.crypto` package installed as part of the IBM Storage Scale software installation cannot mount the entire file system. This is true even if the encryption is applied only for a fileset within the file system being mounted and not for the whole file system.
- After the encryption is enabled for a file system or a fileset within the file system, the file system, attribute **encryption** will be changed to yes. This change is permanent and the attribute cannot be changed to no again.
- Any file that exists on a file system when the encryption is enabled for that file system or a fileset will NOT automatically be encrypted.
- Any file created or modified after the encryption has been enabled for a file system or for a fileset that is encrypted by the policy rules becomes encrypted.
- When a new file is created on a location that has encryption enabled, the file has its attribute **Encrypted** set to yes. To access an encrypted file, the client accessing the file must have access to the encryption key associated with that file.

Contents of an encrypted fileset and folder can be listed by a client that does not have the required encryption keys available, but any other access to the files is denied.

To determine whether a file is encrypted or not, examine the attributes of the file.

Example 3-57 Checking file attributes using the `mmlsattr` command

```
[root@scale1 ~]# mmlsattr -L /ibm/fs01/Turku/Nuuska_Muikkusen_osoite_etelässä.txt
file name:                /ibm/fs01/Turku/Nuuska_Muikkusen_osoite_etelässä.txt
metadata replication: 2 max 2
data replication:        2 max 2
immutable:               no
appendOnly:              no
flags:
storage pool name:       system
fileset name:            Turku
snapshot name:
creation time:           Thu Sep  7 16:55:49 2023
Misc attributes:         ARCHIVE
Encrypted:               yes
[root@scale1 ~]#
```



GNR encryption and disk hospital

IBM Storage Scale System provides a high throughput data service by using many physical disks together to satisfy encrypted data read/write operations. As there are many disks in each building block, disk failure might occur at a higher frequency than with stand-alone disks. With the disk hospital function, IBM Storage Scale System can diagnose the state of the physical disks when an I/O error occurs and can lock and unlock the drive as it attempts to keep data integrity during the rebuilding process. This function gives the IBM Storage Scale system strong reliability and serviceability and maintains data security by using encryption.

The following topics are presented in this chapter:

- ▶ 4.1, “Enabling encryption on IBM Storage Scale system” on page 66
- ▶ 4.2, “Disabling encryption on the IBM Storage Scale system” on page 67
- ▶ 4.3, “Physical disk diagnosis in an SED enabled system” on page 68
- ▶ 4.4, “Replacing physical disks” on page 68
- ▶ 4.5, “Adding physical disks” on page 68
- ▶ 4.6, “Deleting physical disks” on page 69
- ▶ 4.7, “Migration” on page 69

4.1 Enabling encryption on IBM Storage Scale system

Do the following steps to enable GNR encryption on an IBM Storage Scale:

1. Create a Recovery Group by using the **mmvdisk recoverygroup create** command. After the recovery group is created, wait before creating any vdisk sets or file systems. After self encryption is enabled on all self-encrypting drives (SEDs) in the recovery group, vdisk sets and file systems can be created.
2. Use the **mmvdisk sed verify** command to verify that all drives in a recovery group where SED is enabled, are SED capable drives. See Example 4-1.

Example 4-1 mmvdisk sed verify command output

```
mmvdisk sed verify --recovery-group BB01L
```

Disk name	Recovery group	SED Drive
-----	-----	-----
e1s001	BB01L	Yes
e1s002	BB01L	Yes
e1s003	BB01L	Yes
e1s004	BB01L	Yes
e1s005	BB01L	Yes
e1s006	BB01L	Yes
e1s013	BB01L	Yes
e1s014	BB01L	Yes
e1s015	BB01L	Yes

3. When the output from step 2 confirms all drives in each recovery group are SED capable, use the **mmkeyserv** command to create a key by using a GKLM server.
4. After a key is created with step 3, use the **mmvdisk sed enroll** command to enable encryption on drives in each recovery group.

Example 4-2 mmvdisk sed enroll command

```
mmvdisk sed enroll --recovery-group BB01L --rkmid rkm_sedKeyId --key-uuid  
KEY-86a24d4-13894496-36b6-4688-b638-bfb2698bde39
```

```
mmvdisk: Enrolling disks in recoverygroup rg1_3500_P12N with new key from  
default MSID
```

```
mmvdisk: Verifying the disks of RG BB01L for SED support.
```

```
mmvdisk: Successfully enrolled e1s01 with sedKeyId
```

```
mmvdisk: Successfully enrolled e1s02 with sedKeyId
```

```
mmvdisk: Successfully enrolled e1s03 with sedKeyId
```

```
mmvdisk: Successfully enrolled e1s04 with sedKeyId
```

```
mmvdisk: Successfully enrolled e1s05 with sedKeyId
```

```
mmvdisk: Successfully enrolled e1s06 with sedKeyId
```

```
mmvdisk: Successfully enrolled e1s13 with sedKeyId
```

```
mmvdisk: Successfully enrolled e1s14 with sedKeyId
```

```
mmvdisk: Successfully enrolled e1s15 with sedKeyId
```

```
mmvdisk: Successfully enrolled e1s16 with sedKeyId
```

```
mmvdisk: Successfully enrolled e1s17 with sedKeyId
```

```
mmvdisk: Successfully enrolled e1s18 with sedKeyId
```

5. Confirm that the output of the **mmvdisk sed list** command shows that all drives in each recovery group are enrolled and unlocked.

Example 4-3 *mmvdisk sed list command*

```
mmvdisk sed list --recovery-group BB01L
```

In nodeclass nc2 SED Configured: True

Disk name	Recovery group	EnrolledStatus/LockedStatus
-----	-----	-----
e1s001	BB01L	Enrolled with sedKeyId/Unlocked
e1s002	BB01L	Enrolled with sedKeyId/Unlocked
e1s003	BB01L	Enrolled with sedKeyId/Unlocked
e1s004	BB01L	Enrolled with sedKeyId/Unlocked
e1s005	BB01L	Enrolled with sedKeyId/Unlocked
e1s006	BB01L	Enrolled with sedKeyId/Unlocked
e1s013	BB01L	Enrolled with sedKeyId/Unlocked
e1s014	BB01L	Enrolled with sedKeyId/Unlocked
e1s015	BB01L	Enrolled with sedKeyId/Unlocked
e1s016	BB01L	Enrolled with sedKeyId/Unlocked
e1s017	BB01L	Enrolled with sedKeyId/Unlocked
e1s018	BB01L	Enrolled with sedKeyId/Unlocked

6. To assign a new encryption key to the drives, use the **mmvdisk sed rekey** command to change an existing key, such as one used to enroll the drives in step 4 on page 66, to a new key. The **rekey** command uses a new Master Encryption Key (MEK) from the GKLM server and configures all the SED drives to use the new key as the MEK.

Example 4-4 *mmvdisk sed rekey command*

```
mmvdisk sed rekey --recovery-group BB01L --rkmid rkm_sedKeyId --key-uuid  
KEY-86a24d4-66b6f796-b178-4778-b45e-2745765d6886
```

```
mmvdisk: Reenrolling disks in recoverygroup rg1_3500_P12N with new key  
mmvdisk: Successfully enrolled e1s01 with sedNewKeyId  
mmvdisk: Successfully enrolled e1s02 with sedNewKeyId  
mmvdisk: Successfully enrolled e1s03 with sedNewKeyId  
mmvdisk: Successfully enrolled e1s04 with sedNewKeyId  
mmvdisk: Successfully enrolled e1s05 with sedNewKeyId  
mmvdisk: Successfully enrolled e1s06 with sedNewKeyId  
mmvdisk: Successfully enrolled e1s13 with sedNewKeyId  
mmvdisk: Successfully enrolled e1s14 with sedNewKeyId  
mmvdisk: Successfully enrolled e1s15 with sedNewKeyId  
mmvdisk: Successfully enrolled e1s16 with sedNewKeyId  
mmvdisk: Successfully enrolled e1s17 with sedNewKeyId  
mmvdisk: Successfully enrolled e1s18 with sedNewKeyId
```

4.2 Disabling encryption on the IBM Storage Scale system

After SED is enabled on a recovery group, you can disable it only by deleting that recoverygroup. Deleting the recoverygroup initiates a crypto erase, which results in the erasure of all user data by changing the actual encryption keys on the disks.

4.3 Physical disk diagnosis in an SED enabled system

IBM Storage Scale RAID can detect and determine the health of physical disks automatically. SED maintains a view of the corresponding physical disks (pdisks) states.

The following list describes the typical physical disk states:

- ▶ Diagnosing

When the pdisk reports IO errors, the ESS disk hospital puts the pdisk into diagnosing state and checks the underlying disk drive. If the pdisk is found to have a problem, the pdisk is put into one of the following states: Missing, Dead or Failing.

- ▶ Missing or SEDLocked

Typically, a pdisk goes to this state when a drive has no power because, for example, it was powered off by a user, the power plug was pulled out, or the facility lost power. Any of these can put the pdisk into the missing or locked state. The ESS disk hospital unlocks the drives. If the drive unlock is successful, the pdisk is put into an OK state. If the recovery group finds too many disks in the missing or locked state and determines that it cannot continue the data service, the recovery group attempts to failover to the other I/O node. During drive discovery, disks can be unlocked and then the pdisk is put into the OK state.

- ▶ Drives getting placed in a locked status that cannot be unlocked automatically can cause the drive to go to the missing or locked state.

Note: Use the `mmhealth` command to monitor the missing or locked state of the pdisk that is triggering events or alerts. The output advises the user to determine whether the drive is locked by using the `mmvdisk sed list` command.

4.4 Replacing physical disks

When a pdisk is placed into an error state, the recovery group rebuilds data from it into spare spaces.

When the user needs to replace a physical disk in the disk enclosure, the `mmispdisk --not-ok` command can help find the pdisk that needs to be replaced. After the pdisk is located, the user can run the `mmchcarrier` command to start the replacement procedure. After a drive is replaced with a new drive with the `mmchcarrier` command in the GNR system that is configured for encryption, the new drive is configured to support encryption, too. After a successful configuration, the SED drives are unlocked so that data can be accessed and modified.

4.5 Adding physical disks

New drives are added with the `mmaddpdisk` command. On a GNR system that is configured for encryption, the new drive is configured to support encryption. After the drive is configured, the SED drives are unlocked so that data can be accessed or modified.

4.6 Deleting physical disks

When the pdisks are deleted in an SED configured system, they are crypto erased. The data is no longer accessible. The drives are also unenrolled, so they are no longer configured with the MEK.

When the pdisks are deleted, SED is disabled, and the drives are restored back to their original configuration.

4.7 Migration

Note: SED migration is supported by ESS 6.1.6 or later.

When working with existing recovery groups where a file system is already created, and an active workload is present, SED can be enabled as a migration. The migration can be done while a workload is present on a live system. This migration can be done using the **mmvdisk sed enroll** command. See the steps from 4.1, “Enabling encryption on IBM Storage Scale system” on page 66 sub-steps 2-5. The migration uses an MEK from a GKLM server and configures all the drives of a recovery group to enable SED support.

If for any reason during the migration the enroll process is interrupted, then the same command can be reissued to resume the migration. All the drives of the recovery group must be in the OK state to migrate the drives of a recovery group to use SED support.



REDP-5707-00

ISBN 0738421189

Printed in U.S.A.

Get connected

