THALES

Thales Luna Network HSM 7

CLUSTER ADMINISTRATION GUIDE

Document Information

Last Updated	2023-05-25 23:37:30 GMT-04:00

Disclaimer
All information herein is either public information or is the property of and owned solely by Thales Group and/or its subsidiaries who shall have and keep the sole right to file patent applications or any other kind of intellectual property protection in connection with such information.

Nothing herein shall be construed as implying or granting to you any rights, by license, grant or otherwise, under any intellectual and/or industrial property rights of or concerning any of Thales Group's information.

This document can be used for informational, non-commercial, internal, and personal use only provided that:

- The copyright notice, the confidentiality and proprietary legend and this full warning notice appear in all copies.
- This document shall not be posted on any publicly accessible network computer or broadcast in any media, and no modification of any part of this document shall be made.

Use for any other purpose is expressly prohibited and may result in severe civil and criminal liabilities.

The information contained in this document is provided "AS IS" without any warranty of any kind. Unless otherwise expressly agreed in writing, Thales Group makes no warranty as to the value or accuracy of information contained herein.

The document could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Furthermore, Thales Group reserves the right to make any change or improvement in the specifications data, information, and the like described herein, at any time.

Thales Group hereby disclaims all warranties and conditions with regard to the information contained herein, including all implied warranties of merchantability, fitness for a particular purpose, title and non-infringement. In no event shall Thales Group be liable, whether in contract, tort or otherwise, for any indirect, special or consequential damages or any damages whatsoever including but not limited to damages resulting from loss of use, data, profits, revenues, or customers, arising out of or in connection with the use or performance of information contained in this document.

Thales Group does not and shall not warrant that this product will be resistant to all possible attacks and shall not incur, and disclaims, any liability in this respect. Even if each product is compliant with current security standards in force on the date of their design, security mechanisms' resistance necessarily evolves according to the state of the art in security and notably under the emergence of new attacks. Under no circumstances, shall Thales Group be held liable for any third party actions and in particular in case of any successful attack against systems or equipment incorporating Thales products. Thales Group disclaims any liability with respect to security for direct, indirect, incidental or consequential damages that result from any use of its products. It is further stressed

that independent testing and verification by the person using the product is particularly encouraged, especially in any application in which defective, incorrect or insecure functioning could result in damage to persons or property, denial of service, or loss of privacy.

Regulatory Compliance

This product complies with the following regulatory regulations. To ensure compliancy, ensure that you install the products as specified in the installation instructions and use only Thales-supplied or approved accessories.

USA, FCC

This equipment has been tested and found to comply with the limits for a “Class B” digital device, pursuant to part 15 of the FCC rules.

Canada

This class B digital apparatus meets all requirements of the Canadian interference-causing equipment regulations.

Europe

This product is in conformity with the protection requirements of EC Council Directive 2014/30/EU. This product satisfies the CLASS B limits of EN55032.

CONTENTS

PREFACE: About the Cluster Administration Guide

This document describes the setup and administrative procedures required to use clusters and keyrings on your Luna Network HSM.

> **NOTE** This feature requires Luna Network HSM Appliance Software 7.8.0 or newer, Luna HSM Firmware 7.8.0 or newer, and each appliance in the cluster must have the **cluster** secure package installed. Luna HSM Client 10.5.0 or newer is required to access keyrings on a cluster. This feature is currently available on password-authenticated Luna Network HSMs only.

The preface includes the following information about this document:

> "Support Contacts" on page 9

For information regarding the document status and revision history, see "Document Information" on page 2.

Customer Release Notes

The Customer Release Notes (CRN) provide important information about specific releases. Read the CRN to fully understand the capabilities, limitations, and known issues for each release. You can view the latest version of the CRN at www.thalesdocs.com.

Audience

This document is intended for personnel responsible for maintaining your organization's security infrastructure. This includes Luna HSM users and security officers, key manager administrators, and network administrators.

All products manufactured and distributed by Thales are designed to be installed, operated, and maintained by personnel who have the knowledge, training, and qualifications required to safely perform the tasks assigned to them. The information, processes, and procedures contained in this document are intended for use by trained and qualified personnel only.

It is assumed that the users of this document are proficient with security concepts.

Document Conventions

This document uses standard conventions for describing the user interface and for alerting you to important information.

Notes

Notes are used to alert you to important or helpful information. They use the following format:

> **NOTE** Take note. Contains important or helpful information.

Cautions

Cautions are used to alert you to important information that may help prevent unexpected results or data loss. They use the following format:

> **CAUTION!** Exercise caution. Contains important information that may help prevent unexpected results or data loss.

Warnings

Warnings are used to alert you to the potential for catastrophic data loss or personal injury. They use the following format:

> ****WARNING** Be extremely careful and obey all safety and security measures. In this situation you might do something that could result in catastrophic data loss or personal injury.**

Command syntax and typeface conventions

Format	Convention
bold	The bold attribute is used to indicate the following: > Command-line commands and options (Type **dir /p**.) > Button names (Click **Save As**.) > Check box and radio button names (Select the **Print Duplex** check box.) > Dialog box titles (On the **Protect Document** dialog box, click **Yes**.) > Field names (**User Name**: Enter the name of the user.) > Menu names (On the **File** menu, click **Save**.) (Click **Menu** > **Go To** > **Folders**.) > User input (In the **Date** box, type **April 1**.)
italics	In type, the italic attribute is used for emphasis or to indicate a related document. (See the *Installation Guide* for more information.)
<variable>	In command descriptions, angle brackets represent variables. You must substitute a value for command line arguments that are enclosed in angle brackets.
[**optional**] [<optional>]	Represent optional **keywords** or <variables> in a command line description. Optionally enter the keyword or <variable> that is enclosed in square brackets, if it is necessary or desirable to complete the task.
{**a**|**b**|**c**} {<a>|<b>|<c>}	Represent required alternate **keywords** or <variables> in a command line description. You must choose one command line argument enclosed within the braces. Choices are separated by vertical (OR) bars.
[**a**|**b**|**c**] [<a>|<b>|<c>]	Represent optional alternate keywords or variables in a command line description. Choose one command line argument enclosed within the braces, if desired. Choices are separated by vertical (OR) bars.

Support Contacts

If you encounter a problem while installing, registering, or operating this product, please refer to the documentation before contacting support. If you cannot resolve the issue, contact your supplier or Thales Customer Support. Thales Customer Support operates 24 hours a day, 7 days a week. Your level of access is governed by the support plan negotiated between Thales and your organization. Please consult this plan for details regarding your entitlements, including the hours when telephone support is available to you.

Customer Support Portal

The Customer Support Portal, at https://supportportal.thalesgroup.com, is where you can find solutions for most common problems and create and manage support cases. It offers a comprehensive, fully searchable database of support resources, including software and firmware downloads, release notes listing known problems and workarounds, a knowledge base, FAQs, product documentation, technical notes, and more.

> **NOTE** You require an account to access the Customer Support Portal. To create a new account, go to the portal and click on the **REGISTER** link.

Telephone

The support portal also lists telephone numbers for voice contact (Contact Us).

CHAPTER 1: Clusters and Keyrings

Luna Network HSM now allows you to store your cryptographic objects in an encrypted *cluster* within the appliance memory. This process uses Scalable Key Storage (SKS) to encrypt the cluster and the SMK is shared with all member HSMs. The cluster contains *keyrings*, which are analogous to application partitions and can be accessed by a client in much the same way, by connecting to any member appliance. Keys are retrieved from the cluster, decrypted within the secure confines of the HSM, and used by the HSM for cryptographic operations. This configuration allows you to store many more keys than you can normally store within HSM partitions. The management of backup and restore operations is greatly simplified; the HSM administrator can back up the full content of a cluster, at scheduled intervals or on demand.

A cluster can consist of one Luna Network HSM member appliance, or multiple appliances that share the contents of the cluster. Adding multiple members to a cluster improves performance, and provides redundancy and failover for your client applications. Thales recommends a maximum of 4 members per cluster.

Up to 3000 keyrings can be created on the cluster, and each keyring can contain up to 256 objects.

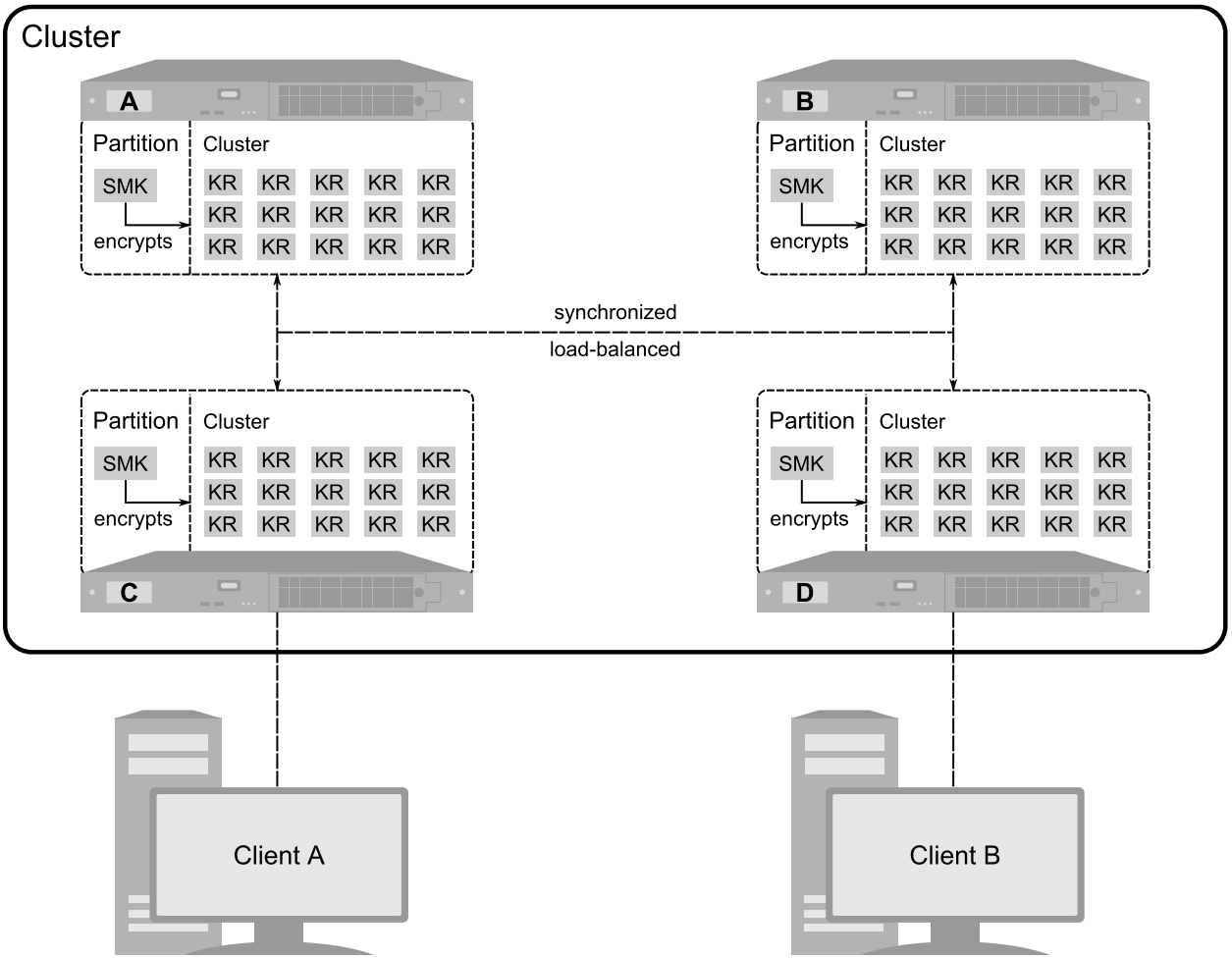
CAUTION! ***TECHNICAL PREVIEW -- EVALUATION ENVIRONMENT ONLY***

Clusters and keyrings are presented as a technical preview, to give customers the opportunity to validate our new HSM management features, designed to reduce operation cost and maximize the return on investment of a fleet of HSMs. This release does not provide a migration path from standard Luna partitions or Luna Cloud HSM services to keyrings. This preview is currently available on password-authenticated Luna Network HSMs only.

DO NOT INSTALL THE CLUSTER PACKAGE ON A LUNA NETWORK HSM IN PRODUCTION

When the **cluster** package is installed, access to any existing partitions on the HSM is disabled, and this can only be reversed by re-imaging the Luna Network HSM appliance (see Re-Imaging the Appliance to Factory Baseline). Re-imaging is a destructive action; all roles, partitions, and keys are destroyed. The Luna Network HSM must be completely reconfigured; all partitions must be recreated and their contents restored from backup. In particular, do not attempt to configure clustering on a Luna Network HSM that already has V1 partitions created; either delete these partitions or re-image the appliance before configuring a cluster.

NOTE This feature requires Luna Network HSM Appliance Software 7.8.0 or newer, Luna HSM Firmware 7.8.0 or newer, and each appliance in the cluster must have the **cluster** secure package installed. Luna HSM Client 10.5.0 or newer is required to access keyrings on a cluster. This feature is currently available on password-authenticated Luna Network HSMs only.



This section will guide you through key concepts and procedures required to set up, manage, and use your Luna Network HSM cluster.

The SKS Master Key

Objects on the cluster are encrypted by a master key, which is created and stored on a V1 application partition on the primary cluster member. New members that join the cluster import the same SMK to a V1 partition on their local HSM. This allows each member to have read-write access to objects on the shared cluster. Refer to Scalable Key Storage (SKS) for more information.

Client-Cluster Connections

Each Luna HSM Client that runs applications using objects on the cluster directs its traffic to a specified cluster member. The **LNHClientRegistration** script is provided with the client software for this purpose. The member specified when running the client registration script acts as the first point of contact between that client and the cluster; all client application requests are directed to that member, and then load-balanced to other cluster members.

You require Luna HSM Client 10.5.0 or newer to connect to a cluster. Refer to "Cluster-Client Connections" on page 33 for guidelines on using the **LNHClientRegistration** script.

Synchronization and Load-Balancing

Keyrings and their contents are synchronized across all members of the cluster, so any member can be queried by client applications for cryptographic operations. Appliance user accounts are also synchronized via the cluster, so users with **admin**, **operator**, and **monitor** privileges can log in to any member. In a cluster with two or more members, the users/roles configuration stored in the cluster are taken as the authority -- if an appliance with custom users/roles joins the cluster, they are overwritten by the users/roles stored in the cluster. This ensures that all cluster members have the same authorized users, and that those users can log in to any individual cluster member.

Client operations are load-balanced across the cluster members to optimize performance and availability. Load-balancing can be customized by moving members between "Affinity Groups" on the next page as described below.

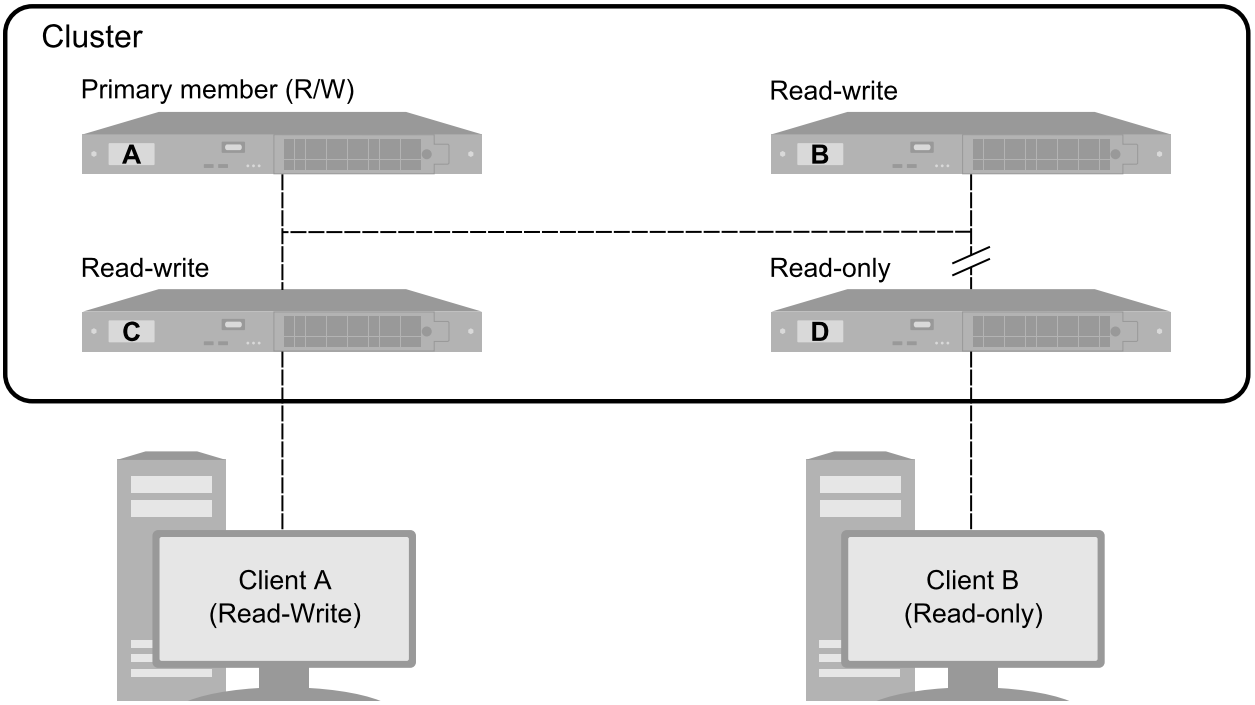
The Primary Member

The Luna Network HSM appliance where the cluster is created becomes the *primary* cluster member. The primary member always has read-write privileges to the cluster; other members have read-write privileges as long as they maintain a network connection to the primary member. If a member's connection to the primary member is disrupted, that member becomes read-only until the connection is re-established. New and existing client sessions become read-only. This applies to connections between the primary and the other members, and is not affected by the client; if a member becomes read-only, the client will not fail over to another member for operations requiring write privileges; these operations will fail. This is necessary to prevent objects stored on the cluster from becoming de-synchronized between members.

You can manually promote a member to primary at any time, as long as that member has read-write privileges at the time of promotion. If the primary cluster member loses connectivity to the cluster, all other members become read-only until it is reconnected. If the primary member is unrecoverable, you must manually remove it from the cluster, at which time another member will automatically be promoted to primary, and the cluster members regain read-write privileges.

See "Promoting a Member to Primary" on page 29.

In the example below, member D has lost connectivity to the primary cluster member. Thus, Client B can perform only operations that do not require write privileges, until member D re-establishes a connection to the primary member, or Client B's traffic is directed to a different member.

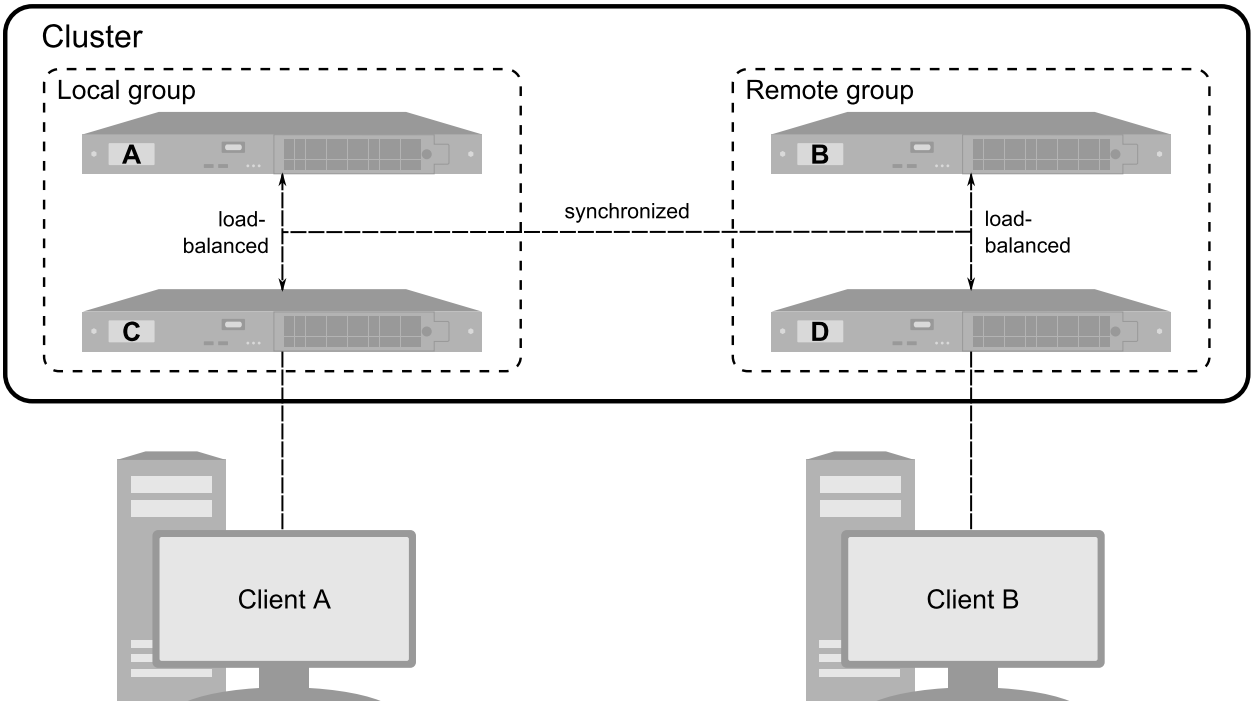


Affinity Groups

Luna Network HSMs within a cluster can be configured in one of two affinity groups (**local** and **remote**), meaning that operations from a connected client application are load-balanced between members of the same group only. This allows you to use the other members, which can be at a remote location with greater latency, as backup or standby members for a specific client. If all members of a client's preferred group are unavailable, operations then fail over to the other group. The state of keyrings and objects stored on them is always synchronized across all members of the cluster, regardless of group. When one or more members of the preferred group become available again, you must restart your client application to move operations back to the preferred group.

New members are added to the **local** group by default and can be moved to the **remote** group (see "Moving a Member to a Different Group" on page 28).

In the example below, the groups are configured so that Client A sends operation requests to cluster member C, (which are load-balanced between members A and C) and Client B sends operation requests to cluster member D (which are load-balanced between members B and D). Each group acts as a standby group for the other.



Keyring Roles and Authentication

Each keyring has two roles that are analogous to the Partition Security Officer and Crypto Officer roles on a standard Luna partition. They are referred to here as:

- **Keyring Security Officer (KRSO)**: initially set by the Partition Security Officer for the partition that created the cluster
- **Keyring Crypto Officer (KRCO)**: performs cryptographic operations on the keyring

Unlike standard Luna partitions, role separation between the KRSO and KRCO roles is not enforced. Separation is enforced, however, between the keyring roles and the cluster security officer (PO of the partition where the cluster's SMK is stored).

Keyring Object Attributes

Keyrings can be used much like standard Luna partitions to create and store cryptographic objects, and perform operations using those objects. The following attributes may be set on keyring objects:

- CKA_LABEL
- CKA_ECDSA_PARAMS
- CKA_EC_POINT
- CKA_TOKEN
- CKA_VALUE
- CKA_KEY_TYPE
- CKA_CLASS
- CKA_UNWRAP

- CKA_SIGN
- CKA_DECRYPT
- CKA_ID
- CKA_MODULUS
- CKA_WRAP
- CKA_PUBLIC_EXPONENT

Cluster Backup/Restore

When Luna Network HSM is configured as a cluster, the entire contents of the cluster can be backed up to the appliance in an encrypted file, accessible to the **admin** user. You can perform backups on demand, or schedule periodic backups and determine how many to store before the oldest ones are overwritten. You can restore the entire cluster from a backup at any time. See "Cluster Backup and Restore" on page 40 for procedures.

CHAPTER 2: Installing or Updating the Cluster Package

Use these instructions to install or update the **cluster** secure package. You must use LunaSH to perform the updates.

> **CAUTION!** ***TECHNICAL PREVIEW -- EVALUATION ENVIRONMENT ONLY***
>
> Clusters and keyrings are presented as a technical preview, to give customers the opportunity to validate our new HSM management features, designed to reduce operation cost and maximize the return on investment of a fleet of HSMs. This release does not provide a migration path from standard Luna partitions or Luna Cloud HSM services to keyrings. This preview is currently available on password-authenticated Luna Network HSMs only.
>
> ***DO NOT INSTALL THE CLUSTER PACKAGE ON A LUNA NETWORK HSM IN PRODUCTION***
>
> When the **cluster** package is installed, access to any existing partitions on the HSM is disabled, and this can only be reversed by re-imaging the Luna Network HSM appliance (see Re-Imaging the Appliance to Factory Baseline). Re-imaging is a destructive action; all roles, partitions, and keys are destroyed. The Luna Network HSM must be completely reconfigured; all partitions must be recreated and their contents restored from backup. In particular, do not attempt to configure clustering on a Luna Network HSM that already has V1 partitions created; either delete these partitions or re-image the appliance before configuring a cluster.

Prerequisites

- The Luna Network HSM must be configured and accessible over the network.
- The Luna Network HSM must be initialized (see Initializing the HSM).
- You require a client operating system supported by Luna HSM Client 10.5.0 or newer.
- You must have Luna Network HSM Appliance Software 7.8.0 or newer (see Updating the Luna Network HSM Appliance Software) and Luna HSM Firmware 7.8.0 or newer (see Updating the Luna HSM Firmware) installed.

To install the cluster package on the Luna Network HSM

1. Transfer the secure package update file to the Luna Network HSM using **pscp** or **scp**.

   pscp <path>**/lnh_cluster-1.0.#-###.spkg admin@**<appliance_host/IP>**:**
2. Using a serial or SSH connection, log in to the appliance as **admin** (see Logging In to LunaSH).
3. Log in as HSM SO (see Logging In as HSM Security Officer).

   lunash:> **hsm login**
4. [Optional] Verify that the secure package file is present on the Luna Network HSM.

lunash:> **package listfile**

5. [Optional] Verify the package file, specifying the authorization code you received from Thales.

   lunash:> **package verify** <filename>**.spkg -authcode** <code_string>

6. Install the secure package for the **cluster** service.

   lunash:> **package update lnh_cluster-1.0.#-###.spkg -authcode** <authcode_string>

7. Run the restart command for the **cluster** service, as prompted. This step completes the service installation procedure.

   lunash:> **service restart cluster**

   The new **cluster** and **keyring** commands become available when you open a new LunaSH session.

8. If you plan to use REST API to work with clusters, set up the **webserver** service so that the appliance can accept calls from your web application.

   Refer to Webserver Setup.

   After configuring the **webserver** service, you must synchronize the HSM time with the time on the appliance.

   lunash:> **hsm time sync**

9. Install the Luna HSM Client software on the client machine you will use to configure your application partition (s).

   Refer to Luna HSM Client Software Installation.

Troubleshooting

If you encounter any issues, refer to Reading System Logs to check recent activity on the appliance. To report an issue that is not described below, export the appliance syslog to a client workstation and provide it to your Thales representative (refer to Exporting System Logs).

CHAPTER 3: Configuring the Luna Network HSM for Clustering

The following procedures will allow you to set up clustering on the Luna Network HSM, using "REST API" below calls or "LunaSH" on page 20. This procedure must be completed whether you are creating a new cluster (see "Creating a Cluster" on page 23), or joining an existing cluster (see "Adding a New Member to an Existing Cluster" on page 25).

> **NOTE** This feature requires Luna Network HSM Appliance Software 7.8.0 or newer, Luna HSM Firmware 7.8.0 or newer, and each appliance in the cluster must have the **cluster** secure package installed. Luna HSM Client 10.5.0 or newer is required to access keyrings on a cluster. This feature is currently available on password-authenticated Luna Network HSMs only.

> **CAUTION!** Only one cluster can exist on the appliance. Do not attempt to:
> - create multiple application partitions on the HSM
> - create more than one cluster on the appliance
> - join a second cluster when the appliance is already a member of a different cluster

REST API

Prerequisites

- The Luna Network HSM must be configured and accessible over the network.
- The Luna Network HSM must be initialized.

  PUT /api/lunasa/hsms/{hsmid}
- The Luna Network HSM and any clients must have NTP configured, or have their system clocks synchronized manually. Clocks must be synchronized before starting the cluster service:
  - **Using NTP** (actionid: **synchronize**)**:** POST /api/lunasa/ntp/actions/{actionid}
  - **Manually:** PUT /api/lunasa/time

> **CAUTION!** If the system clock is adjusted after the cluster certificate is created, the certificates might not be valid due to date/time. For example, if the certificate is generated while the system clock is ahead by a few minutes, and the clock is then corrected, the certificate will not be valid until the clock catches up to the time it was set to when the cert was created. If the current system time does not fall within the certificate's range of validity, the **cluster** service fails to start. Refer to known issue LUNA-27578.

> The Luna Network HSM must have only a single network interface configured (see Configuring IP and Network Parameters). All members of a cluster must be able to communicate bi-directionally with all other members.

To configure the appliance for clustering

1. Create a new application partition or provision an existing partition for cluster and keyring creation.
   a. Create a new application partition, specifying a V1 partition.

   > **CAUTION!** Using a V0 partition to create a cluster can cause undesired behavior and require you to factory reset the **cluster** service and recreate the partition.
   > Only one V1 partition can exist on an appliance configured for clustering; do not create more than one V1 partition.

   POST /api/lunasa/hsms/{hsmid}/partitions

   b. Initialize the partition and the Partition Security Officer (PO) role.

   > **NOTE** The partition that will hold the SKS Master Key (SMK) on each cluster member must use the same cloning domain. If you are configuring this appliance to join an existing cluster, ensure that you use the same domain string that was used to initialize the partition on the primary member.

   PUT /api/lunasa/hsms/{hsmid}/partitions/{partitionid}

   c. Turn off Per-Key Authorization by setting partition policy **40** to **0**.

   PUT /api/lunasa/hsms/{hsmid}/partitions/{partitionid}/policies/{policyid}

   > **CAUTION!** Changing partition policies after you have created and are using clusters/keyrings can cause unwanted behavior. Ensure that you have configured all your desired policy settings before you continue.

   d. Initialize the Crypto Officer (CO) role (roleid: **co**).

   > **NOTE** If **HSM policy 21: Force user PIN change after set/reset** is enabled (this is the default setting), the CO must change the password from its initial value before any other actions are permitted.

      i. PUT /api/lunasa/hsms/{hsmid}/partitions/{partitionid}/roles/{roleid}
      ii. PATCH /api/lunasa/hsms/{hsmid}/partitions/{partitionid}/roles/{roleid}

2. Enable the **cluster** service on the appliance (serviceid: **cluster**, actionid: **enable**). Enabling the service means that it will be automatically restarted in the case of an appliance reboot.

   POST /api/lunasa/services/{serviceid}/actions/{actionid}

3. Start the **cluster** service on the appliance (serviceid: **cluster**, actionid: **start**).

> **NOTE** The **cluster** service cannot be started if a user partition has not yet been created (refer to step 1).

POST /api/lunasa/services/{serviceid}/actions/{actionid}

4. Check the status of the **cluster** service (enabled and running). It may take several minutes for the **cluster** service to start, and the next step will fail if it is still starting up.

   GET /api/lunasa/services/{serviceid}

5. Set the IP address to use for the core cluster traffic on this member appliance. This must be the same IP address as one of the appliance's network interfaces.

   > **NOTE** The core IP address cannot be changed later, except after factory reset. Refer to "Configuring the Luna Network HSM for Clustering" on page 18.

   PUT /api/cluster/config

   ```
   curl -X PUT "https://1.2.3.4:50070/api/cluster/config" --data '{"service": "core",
   "IpAddress": "1.2.3.4"}' -H "Content-Type: application/json" -H "Authorization: Basic
   dXNlcjpwYXNzd29yZA==" --tlsv1.2 --insecure
   ```

   > **NOTE** If you have configured more than one network interface on this Luna Network HSM, Thales recommends restarting the **cluster** service before continuing.

6. [Optional] You can also bind the administrative and cryptographic traffic to the cluster to specified network devices, including assigning a port number from a limited range (**admin**: **50075-50079**, **crypto**: **50055-50059**).

   PUT /api/cluster/config

   ```
   curl -X PUT  "https://1.2.3.4:50070/api/cluster/config" --data '{"service": "crypto",
   "interface": "all", "port": "50055"}' -H "Content-Type: application/json" -H "Authorization:
   Basic dXNlcjpwYXNzd29yZA==" --tlsv1.2 --insecure
   ```

   > **CAUTION!** In this release, changing the default port used for crypto operations on the cluster (**50052**) can cause communication problems between cluster members. Refer to known issue LUNA-26485.

7. [Optional] Check that the network configuration was set correctly.

   GET /api/cluster/config

   ```
   curl "https://1.2.3.4:50070/api/cluster/config"  -H "Content-Type: application/json" -H
   "Authorization: Basic dXNlcjpwYXNzd29yZA==" --insecure
   ```

LunaSH

Prerequisites

> The Luna Network HSM must be configured and accessible over the network.

> The Luna Network HSM must be initialized (see Initializing the HSM).

- **HSM policy 21: Force user PIN change after set/reset** must be set to **OFF** (see Setting HSM Policies Manually).
- The Luna Network HSM and any clients must have NTP configured, or have their system clocks synchronized manually. Clocks must be synchronized before starting the **cluster** service. Refer to NTP on Luna Network HSM or Setting the System Date and Time.

> **CAUTION!** If the system clock is adjusted after the cluster certificate is created, the certificates might not be valid due to date/time. For example, if the certificate is generated while the system clock is ahead by a few minutes, and the clock is then corrected, the certificate will not be valid until the clock catches up to the time it was set to when the cert was created. If the current system time does not fall within the certificate's range of validity, the **cluster** service fails to start. Refer to known issue LUNA-27578.

To configure a partition for clustering

1. Create a new application partition for cluster and keyring creation.
   a. Create a new application partition in LunaSH, specifying a V1 partition.

   > **CAUTION!** Using a V0 partition to create a cluster can cause undesired behavior and require you to factory reset the **cluster** service and recreate the partition.
   >
   > Only one V1 partition can exist on an appliance configured for clustering; do not create more than one V1 partition.

   Refer to Creating or Deleting an Application Partition.

   b. Create a template file to initialize the partition with your desired policy settings. **Partition policy 40: Require Per-Key Authorization Data** must be set to **0**. You must create the file yourself -- it is not possible to export the default template file from the partition using LunaSH.

   Refer to Editing a Partition Policy Template. Use **pscp**/**scp** to transfer the policy file to the appliance filesystem. Your template file must contain the following line:

   ```
   40:"Require Per-Key Authorization Data":0:1:0
   ```

   > **CAUTION!** Changing partition policies after you have created and are using keyrings can cause unwanted behavior. Ensure that you have configured all your desired policy settings before you continue.

   c. Initialize the partition and the Partition Security Officer (PO) role in LunaSH, specifying the policy template filename.

   > **NOTE** The partition that will hold the SKS Master Key (SMK) on each cluster member must use the same cloning domain. If you are configuring this appliance to join an existing cluster, ensure that you use the same domain string that was used to initialize the partition on the primary member.

   Refer to Initializing an Application Partition in the product documentation.

   d. Initialize the Crypto Officer (CO) role in LunaSH.

Refer to Initializing the Crypto Officer Role.

2. Start the **cluster** service on the appliance, or, if you want the service to be started automatically on appliance reboot, enable the **cluster** service on the appliance.

   > **NOTE** The **cluster** service cannot be started if a user partition has not yet been created (refer to step 1).

   - **To enable the cluster service on the appliance (optional):**

     lunash:> **cluster enable**

     The **cluster** service starts automatically after being enabled.

   - **To start the cluster service without enabling it:**

     lunash:> **service start cluster**

3. Check the status of the **cluster** service (enabled and/or running).

   > **NOTE** It can take up to 1-2 minutes for the **cluster** service to start. Until startup is complete, the status is reported as **starting**.

   lunash:> **cluster status**

4. Set the IP address to use for the core cluster traffic on this member appliance. This must be the same IP address as one of the appliance's network interfaces.

   > **NOTE** The core IP address cannot be changed later, except after factory reset. Refer to "Configuring the Luna Network HSM for Clustering" on page 18.

   lunash:> **cluster config -service core -ipaddress** <appliance_IP>

5. [Optional] You can also bind the administrative and cryptographic traffic to the cluster to specified network devices, including assigning a port number from a limited range (**admin**: **50075-50079**, **crypto**: **50055-50059**).

   lunash:> **cluster config -service admin** [**-interface** <netdevice>] [**-port** <port>]

   lunash:> **cluster config -service crypto** [**-interface** <netdevice>] [**-port** <port>]

   > **CAUTION!** In this release, changing the default port used for crypto operations on the cluster (**50052**) can cause communication problems between cluster members. Refer to known issue LUNA-26485.

6. [Optional] Check that the network configuration was set correctly.

   lunash:> **cluster config show**

CHAPTER 4: Managing Cluster Members

These procedures will allow you to add new members to an existing cluster, remove a functioning member, or delete a non-functioning member from the cluster configuration:

> **NOTE** This feature requires Luna Network HSM Appliance Software 7.8.0 or newer, Luna HSM Firmware 7.8.0 or newer, and each appliance in the cluster must have the **cluster** secure package installed. Luna HSM Client 10.5.0 or newer is required to access keyrings on a cluster. This feature is currently available on password-authenticated Luna Network HSMs only.

Creating a Cluster

The following procedures will allow you to create a new cluster on the Luna Network HSM appliance, and authorize the local member on the appliance to create objects on the cluster, using "REST API" below calls or "LunaSH" on the next page.

> **NOTE** This procedure applies to the first cluster member only, or to a cluster of one appliance. The member on which the cluster is created is the *primary* member, and this affects the behavior of the cluster as a whole. Refer to "The Primary Member" on page 12 for more information. If you want to join the appliance to an existing cluster, see "Adding a New Member to an Existing Cluster" on page 25.

REST API

To create a cluster and authorize the first member

1. Create a cluster on the appliance. You must specify the HSM SO password and a label for the new cluster.

   POST /api/clusters

   ```
   curl -X POST "https://1.2.3.4:50070/api/clusters" --data '{"authentication":[{"type":
   "SO","password": "iamtheHSMSO"}],"clusterLabel": "cluster01"}' -H "Content-Type:
   application/json" -H "Authorization: Basic dXNlcjpwYXNzd29yZA==" --insecure
   ```

2. A default member is created on the cluster with the name **LNH-<HSM_SN>**. Display a list of members on the cluster to find its Member ID.

GET /api/clusters/{clusterid}/members

```
curl "https://1.2.3.4:50070/api/clusters/d533441d-9c3f-403b-9280-6fc6ccaae338/members"  -H
"Content-Type: application/json"  -H "Authorization: Basic dXNlcjpwYXNzd29yZA==" --insecure
```

3. Authorize the member to create objects on the cluster's keyrings by specifying the application partition, and CO credential.

   PUT /api/clusters/{clusterid}/members/{memberid}

   ```
   curl -X PUT "https://1.2.3.4:50070/api/clusters/d533441d-9c3f-403b-9280-
   6fc6ccaae338/members/cf6d105d-a5eb-4d2f-be57-d23b72c9edea" --data '{"authentication":
   [{"type": "CO","partitionLabel": "clusterpar","password":
   "iamtheCO"}],"action":"authorize"}' -H "Content-Type: application/json" -H "Authorization:
   Basic dXNlcjpwYXNzd29yZA==" --insecure
   ```

   > **TIP** Manual member authorization is required only:
   > - to authorize the first member of a cluster
   > - when recovering a cluster with a single member that has failed
   > - when manually reinstating the first member of a cluster where *all* members failed
   >
   > In a cluster where at least one member remains active and connected, rejoining members automatically take their authorization from that member, and manual re-authorization is not needed.

   You can now create new keyrings on the cluster (see "Managing Keyrings" on page 36).

4. You must run the provided **LNHClientRegistration** script on any Luna HSM Client computers that will query this member to create or use keyrings on the cluster. See "Cluster-Client Connections" on page 33 for the procedure.

LunaSH

To create a cluster and authorize the first member

1. Create a cluster on the appliance. You must specify the HSM SO password and a label for the new cluster. Do not create more than one cluster.

   lunash:> **cluster create -label** <label> **-password** <HSMSO_password>

2. A default member is created on the cluster with the name **LNH-<HSM_SN>**. Display a list of members on the cluster to find its Member ID.

   lunash:> **cluster member list -clusterid** <UUID>

3. Authorize the member to create objects on the cluster's keyrings by specifying the application partition, CO credential, and Member ID.

   lunash:> **cluster member authorize -partition** <label> **-copassword** <password> **-memberid** <string>

> **TIP** Manual member authorization is required only:
> > to authorize the first member of a cluster
> > when recovering a cluster with a single member that has failed
> > when manually reinstating the first member of a cluster where *all* members failed
>
> In a cluster where at least one member remains active and connected, rejoining members automatically take their authorization from that member, and manual re-authorization is not needed.

You can now create new keyrings on the cluster (see "Managing Keyrings" on page 36).

4. You must run the provided **LNHClientRegistration** script on any Luna HSM Client computers that will query this member to create or use keyrings on the cluster. See "Cluster-Client Connections" on page 33 for the procedure.

Adding a New Member to an Existing Cluster

Use this procedure to connect a Luna Network HSM appliance to an existing cluster created on another appliance using "REST API" below or "LunaSH" on page 27.

REST API

Prerequisites

> On Appliance A (where the cluster is created):
- The appliance must be configured for clustering (see "Configuring the Luna Network HSM for Clustering" on page 18).
- A cluster must be created on the appliance, and the appliance must be authorized as a member (see "Creating a Cluster" on page 23).
- Thales recommends that new members join the primary cluster member. If there are already multiple members in the cluster, and you are not sure which is currently the primary member, use (the primary member returns `"primarynode": true`):

  GET /api/clusters/{clusterid}/members/{memberid}

> On Appliance B (that will join the existing cluster):
- The appliance must be configured for clustering (see "Configuring the Luna Network HSM for Clustering" on page 18).

> **NOTE** If a cluster already exists on appliance B, it will not be allowed to join appliance A's cluster.

> Both appliances must be accessible to each other over the network. All members of a cluster must be able to communicate bi-directionally with all other members.

> Both appliances should have NTP configured, or the appliance clocks must be synchronized manually; if the appliance times are out of sync by more than 30 seconds, the new member will not be allowed to join the cluster:

Using NTP (actionid: **synchronize**): POST /api/lunasa/ntp/actions/{actionid}

Manually: PUT /api/lunasa/time

> **CAUTION!** You must ensure that the appliance clocks of all your cluster members are synchronized regularly -- if a member appliance's clock drifts further than 30 seconds from the rest of the cluster, that member will not be automatically recovered after a disconnection or reboot.
>
> Ensure that you add only one member to the cluster at a time. Undesired behavior can result from adding more than one member simultaneously.

To join an existing cluster

1. On appliance A, get the cluster ID you wish to join.

   GET /api/clusters

   ```
   curl "https://1.2.3.4:50070/api/clusters/" -H "Content-Type: application/json" -H
   "Authorization: Basic dXNlcjpwYXNzd29yZA==" --tlsv1.2 --insecure

   {"data":[{"clusterLabel":"cluster00","clusterUUID":"2afac9b6-f74c-4888-a059-da015240cd1b"}]}
   ```

2. On appliance B, join appliance A's cluster by specifying its cluster ID, IP address, **admin** user password, the label of the partition on appliance B that will authorize appliance B as a member, and that partition's CO password.

   PUT /api/clusters/{clusterid}

   ```
   curl "https://5.6.7.8:50070/api/clusters/2afac9b6-f74c-4888-a059-da015240cd1b" -X PUT -H
   "Authorization: Basic dXNlcjpwYXNzd29yZA==" -H "Content-Type: application/json" --insecure -
   -tlsv1.2' --data {"action": "join","remoteClusterIpAddress":
   "1.2.3.4","remoteAdminPassword": "IamtheAdmin","authentication":[{"type":
   "CO","partitionLabel": "joiningpar","password": "IamtheCO"}]}'
   ```

   > **NOTE** This operation may take a few minutes, even though a success message is received right away.

   New members joining an existing cluster are authorized automatically.

3. Confirm that both cluster members are now listed. If they are not, wait a few minutes and try again -- the join process may still be underway.

   GET /api/clusters/{clusterid}/members

   ```
   curl "https://5.6.7.8:50070/api/clusters/2afac9b6-f74c-4888-a059-da015240cd1b/members"  -H
   "Content-Type: application/json"  -H "Authorization: Basic dXNlcjpwYXNzd29yZA==" --insecure
   ```

4. You must run the provided **LNHClientRegistration** script on any Luna HSM Client computers that will query this member to create or use keyrings on the cluster. See "Cluster-Client Connections" on page 33 for the procedure.

LunaSH

Prerequisites

> On Appliance A (where the cluster is created):

- The appliance must be configured for clustering (see "Configuring the Luna Network HSM for Clustering" on page 18).
- A cluster must be created on the appliance, and the appliance must be authorized as a member (see "Creating a Cluster" on page 23).
- Thales recommends that new members join the primary cluster member. If there are already multiple members in the cluster, and you are not sure which is currently the primary member, use (the primary member is marked with a "**P**"):

  lunash:> **cluster member list -clusterid** <UUID>

> On Appliance B (that will join the existing cluster):

- The appliance must be configured for clustering (see "Configuring the Luna Network HSM for Clustering" on page 18).

> **NOTE** If a cluster already exists on appliance B, it will not be allowed to join appliance A's cluster.

> Both appliances must be accessible to each other over the network. All members of a cluster must be able to communicate bi-directionally with all other members.

> Both appliances should have NTP configured, or the appliance clocks must be synchronized manually; if the appliance times are out of sync by more than 30 seconds, the new member will not be allowed to join the cluster. See Setting the System Date and Time.

> **CAUTION!** You must ensure that the appliance clocks of all your cluster members are synchronized regularly -- if a member appliance's clock drifts further than 30 seconds from the rest of the cluster, that member will not be automatically recovered after a disconnection or reboot.
>
> Ensure that you add only one member to the cluster at a time. Undesired behavior can result from adding more than one member simultaneously.

To join an existing cluster

1. On appliance A, get the cluster ID you wish to join.

   lunash:> **cluster list**

2. On appliance B, join appliance A's cluster by specifying its cluster ID, IP address, **admin** user password, the label of the partition on appliance B that will authorize appliance B as a member, and that partition's CO password.

   lunash:> **cluster join -clusterid** <string> **-clusteripaddress** <ipaddress> **-remotepassword** <adminpassword> **-partition** <name> **-copassword** <password> [**-force**]

3. Confirm that both cluster members are now listed. The new member is listed as unauthorized. An asterisk indicates the local member.

   lunash:> **cluster member list -clusterid** <UUID>

   New members joining an existing cluster are authorized automatically.

4. You must run the provided **LNHClientRegistration** script on any Luna HSM Client computers that will query this member to create or use keyrings on the cluster. See "Cluster-Client Connections" on page 33 for the procedure.

Moving a Member to a Different Group

New members are added to the **local** group by default and can be moved to the **remote** group using "REST API" below or "LunaSH" below.

> **NOTE** Members cannot be moved from the **remote** group back to the **local** group using this procedure; you must remove the member from the cluster and then re-add it (see "Removing a Functioning Member From a Cluster" on page 30 and "Adding a New Member to an Existing Cluster" on page 25).

REST API

To move a member to the remote group

1. Ensure the member you wish to move is a member of the **local** group.

   GET /api/clusters/{clusterid}/members

2. Move the member to the **remote** group.

   PUT /api/clusters/{clusterid}/members/{memberid}

```
{
    "action": "config",
    "group": "remote"
}
```

LunaSH

To move a member to the remote group

1. Ensure the member you wish to move is a member of the **local** group (`Member Group: local`).

   lunash:> **cluster member show -memberid** <UUID>

2. Move the member to the **remote** group.

   lunash:> **cluster member config -memberid** <UUID> **-group remote**

3. [Optional] Check the new group configuration.

   lunash:> **cluster member list -clusterid** <UUID>

Promoting a Member to Primary

This procedure allows you to promote any member of the cluster to be the primary member (see "The Primary Member" on page 12 for more information) using "REST API" below or "LunaSH" below. You can promote any active cluster member to primary from any other active member.

REST API

Prerequisites

> The member being promoted must have read-write privileges on the cluster. If you are promoting another member because the original primary member is unrecoverable, you must first delete the original primary from the cluster before you can promote your desired member (see "Deleting an Unrecoverable Member From a Cluster" on the next page).

To promote a member to primary

1. Check that the member you wish to promote has read-write privileges (`"mode": "Read-Write"`).

   GET /api/clusters/{clusterid}/members/{memberid}

2. Promote the member to primary.

   PUT /api/clusters/{clusterid}/members/{memberid}

```
{
    "action": "config",
    "primary": true
}
```

LunaSH

Prerequisites

> The member being promoted must have read-write privileges on the cluster. If you are promoting another member because the original primary member is unrecoverable, you must first delete the original primary from the cluster before you can promote your desired member (see "Deleting an Unrecoverable Member From a Cluster" on the next page).

To promote a member to primary

1. Ensure the member you wish to promote has read-write privileges (`Mode: Read-Write`).

   lunash:> **cluster member show -memberid** <UUID>

2. Promote the member to primary.

   lunash:> **cluster member config -memberid** <UUID> **-primary**

3. [Optional] Check the new primary status (`Primary: True`).

   lunash:> **cluster member show -memberid** <UUID>

Removing a Functioning Member From a Cluster

This procedure allows you to remove a functioning member from the cluster using "REST API" below or "LunaSH" below. This could be done for scheduled maintenance or updates, and assumes that the member is accessible and functioning properly within the cluster. This operation must be done on the member appliance that is leaving the cluster.

> **CAUTION!** Ensure that all other members are online before you proceed. If another member is offline, it will not be updated with the new cluster configuration, and its cluster service may need to be factory reset as a result. Refer to known issue LUNA-25449.

REST API

To remove a member from the cluster

1. Check that the member you wish to remove can communicate with the other cluster members (`"visibleToServicingNode": true`).

   GET /api/clusters/{clusterid}/members/{memberid}

2. Remove the member from the cluster.

   PATCH /api/clusters/{clusterid}/members/{memberid}

```
{
    "authentication":
    [{
        "type": "CO",
        "partitionLabel": "clusterpar",
        "password": "iamtheCO"
    }],
    "action":"leave"
};
```

LunaSH

To remove a member from the cluster

1. Check that the member you wish to remove is the local member (`Local: True`) and can communicate with the other cluster members (`Visible: True`).

   lunash:> **cluster member show -memberid** <UUID>

2. Remove the member from the cluster, specifying the local partition and its Crypto Officer password.

   lunash:> **cluster leave -memberid** <UUID> **-partition** <label> **-copassword** <password>

Deleting an Unrecoverable Member From a Cluster

This procedure allows you to delete a member from the cluster that has become unrecoverable and cannot be accessed via SSH or REST API. This action removes references to the unrecoverable member from the other members of the cluster. This operation must be done on another cluster member using "REST API" on the next

page or "LunaSH" below.

> **CAUTION!** Ensure that all members except those being deleted are online before you proceed. If another member is offline, it will not be updated with the new cluster configuration, and its cluster service may need to be factory reset as a result. Refer to known issue LUNA-25449.

REST API

To delete an unrecoverable member from the cluster

1. Check that the member you wish to remove is not visible to the other cluster members (`"visibleToServicingNode": false`).

   GET /api/clusters/{clusterid}/members/{memberid}

2. Delete the member from the cluster, specifying the Crypto Officer password for the local cluster partition.

   DELETE /api/clusters/{clusterid}/members/{memberid}

```
{
    "authentication":
    [{
        "type": "CO",
        "password": "IamtheCO"
    }]
}
```

> **NOTE** If the member you just removed was the primary member, you must now promote another member to primary (see "Promoting a Member to Primary" on page 29).
>
> When you are able to re-establish an SSH, REST, or serial connection to the deleted member appliance, you must factory reset its **cluster** service before rejoining it to the cluster (serviceid: **config**, actionid: **factoryreset**):
>
> POST /api/lunasa/services/{serviceid}/actions/{actionid}
>
> lunash:> **sysconf config factoryreset -service cluster**

LunaSH

To delete an unrecoverable member from the cluster

1. Check that the member you wish to remove is not visible to the other cluster members (`Visible: False`).

   lunash:> **cluster member show -memberid** <UUID>

2. Delete the member from the cluster, specifying the Crypto Officer password for the local cluster partition.

   lunash:> **cluster member delete -memberid** <UUID> **-copassword** <password>

NOTE If the member you just removed was the primary member, you must now promote another member to primary (see "Promoting a Member to Primary" on page 29).

When you are able to re-establish an SSH, REST, or serial connection to the deleted member appliance, you must factory reset its **cluster** service before rejoining it to the cluster (serviceid: **config**, actionid: **factoryreset**):

POST /api/lunasa/services/{serviceid}/actions/{actionid}

lunash:> **sysconf config factoryreset -service cluster**

CHAPTER 5: Cluster-Client Connections

Thales provides a client-side script, **LNHClientRegistration**, to connect Luna HSM Client to a cluster. You must run the script on any Luna HSM Client computers that will create or use keyrings on the cluster. The client requires a specified member Luna Network HSM to use as an entry point to the cluster. All traffic from the client will be directed to this member appliance, although the operations may be performed by other cluster members.

> **NOTE** This feature requires Luna Network HSM Appliance Software 7.8.0 or newer, Luna HSM Firmware 7.8.0 or newer, and each appliance in the cluster must have the **cluster** secure package installed. Luna HSM Client 10.5.0 or newer is required to access keyrings on a cluster. This feature is currently available on password-authenticated Luna Network HSMs only.

If this client will be used by a customer with **monitor** privileges in a service provider deployment, you can specify a LunaSH username to associate with the client. This can be the default **monitor** role, or a custom user with the **monitor** role assigned (see Creating Custom Appliance User Accounts). The specified user must already exist on the appliance. This feature requires minimum Luna HSM Client 10.5.1 and Luna Network HSM Appliance Software 7.8.1.

Connecting Luna HSM Client to a Cluster

The following procedure will allow you to connect a Luna HSM Client computer to a Luna Network HSM cluster. Each cluster is assigned an index number from 00 to 09 on the client. In this release, the configuration can be viewed only in the **crystoki.ini** / **Chrystoki.conf** configuration file; the following entries for each cluster are added to the **LunaSA Client** section (see Configuration File Summary):

```
LNHServer##
LNHServerClientCert##
LNHServerClientKey##
LNHServerCAFile##
LNHServerCN##
```

Prerequisites

> Ensure that you have the following information about each cluster you want to access using this client:
  - the Cluster ID
  - the IP address for an *authorized* member appliance that will accept this client's traffic

> If you customize the port numbers for admin and/or crypto traffic to the appliance, you must edit the **LNHClientRegistration** script to account for these port numbers, or client registration will fail. To update the script, replace all instances of the default admin port **50070** with your configured admin port, and instances of the default crypto port **50052** with your configured crypto port.

> **CAUTION!** In this release, changing the default port used for crypto operations on the cluster (**50052**) can cause communication problems between cluster members. Refer to known issue LUNA-26485.

To connect Luna HSM Client to a cluster member

1. Run the **LNHClientRegistration** script to connect the client to the cluster, specifying the Cluster ID (**-c**), the IP address of the member the client will connect to (**-i**), a Common Name for the client certificate (**-n**), an optional label for the client (**-l**), and an optional LunaSH username with a **monitor** role to associate with this client (**-u**). This assigns the cluster to the 00 index position on the client.

   - **Linux/AIX:**

     # ./LNHClientRegistration.sh -n <client Common Name> **-i** <IPaddress> **-c** <clusterID> [**-l** <optional_client_label>] [**-u** <monitor_LunaSH_user>]

   - **Windows PowerShell:**

     ./LNHClientRegistration.ps1 -n <client Common Name> **-i** <IPaddress> **-c** <clusterID> [**-l** <optional_client_label>] [**-u** <monitor_LunaSH_user>]

   - **Windows command prompt:**

     powershell.exe -command "LNHClientRegistration.ps1 -n <client Common Name> **-i** <IPaddress> **-c** <clusterID> [**-l** <optional_client_label>] [**-u** <monitor_LunaSH_user>]**"**

2. Run the script again for each additional cluster you wish to add, including the **-m** option to indicate that you are adding multiple clusters. Each new cluster added will be assigned to the next incremental index position (01, 02, 03... 09). If all the positions are filled, an error message is returned.

   > **CAUTION!** Running the script again without the **-m** option will overwrite the cluster configuration at the 00 index position.

   > **NOTE** You require minimum Luna HSM Client 10.5.1 to connect a client to multiple clusters.

   - **Linux/AIX:**

     # ./LNHClientRegistration.sh -n <client Common Name> **-i** <IPaddress> **-c** <clusterID> [**-l** <optional_client_label>] [**-u** <monitor_LunaSH_user>] **-m**

   - **Windows PowerShell:**

     LNHClientRegistration.ps1 -n <client Common Name> **-i** <IPaddress> **-c** <clusterID> [**-l** <optional_client_label>] [**-u** <monitor_LunaSH_user>] **-m**

   - **Windows command prompt:**

     powershell.exe -command "LNHClientRegistration.ps1 -n <client Common Name> **-i** <IPaddress> **-c** <clusterID> [**-l** <optional_client_label>] [**-u** <monitor_LunaSH_user>] **-m"**

Removing a Cluster From the Luna HSM Client

In this release, to remove a cluster from the client, you must manually delete it from the index by editing the LunaSA Client section of the **crystoki.ini** / **Chrystoki.conf** configuration file. Delete the entries for the cluster you wish to remove and save the configuration file.

```
LunaSA Client = {
   ReceiveTimeout = 20000;
   SSLConfigFile = /usr/safenet/lunaclient/bin/openssl.cnf;
   ClientPrivKeyFile = /usr/safenet/lunaclient/cert/client/ClientNameKey.pem;
   ClientCertFile = /usr/safenet/lunaclient/cert/client/ClientNameCert.pem;
   ServerCAFile = /usr/safenet/lunaclient/cert/server/CAFile.pem;
   NetClient = 1;
   TCPKeepAlive = 1;

   LNHServer00 = 1.2.3.4:50052;
   LNHServerClientCert00 = /usr/safenet/lunaclient/cert/client/c2c94c40-6491-409e-bd3d-
16e236544b7f/2.3.4.5.pem;
   LNHServerClientKey00 = /usr/safenet/lunaclient/cert/client/c2c94c40-6491-409e-bd3d-
16e236544b7f/2.3.4.5Key.pem;
   LNHServerCAFile00 = /usr/safenet/lunaclient/cert/server/c2c94c40-6491-409e-bd3d-
16e236544b7f/lnh_ca.pem;
   LNHServerCN00 = lnh.thalesgroup.com;
   LNHServer01 = 5.6.7.8:50052;
   LNHServerClientCert01 = /usr/safenet/lunaclient/cert/client/3fed78e8-58ad-4aec-be5f-
4a12a04ff073/2.3.4.5.pem;
   LNHServerClientKey01 = /usr/safenet/lunaclient/cert/client/3fed78e8-58ad-4aec-be5f-
4a12a04ff073/2.3.4.5Key.pem;
   LNHServerCAFile01 = /usr/safenet/lunaclient/cert/server/3fed78e8-58ad-4aec-be5f-
4a12a04ff073/lnh_ca.pem;
   LNHServerCN01 = lnh.thalesgroup.com;
}
```

CHAPTER 6: Managing Keyrings

The procedures below will allow you to create a new keyring on a cluster, and configure it for use by a Luna HSM Client. Up to 3000 keyrings can be created on the cluster, and each keyring can contain up to 256 objects.

> **NOTE** This feature requires Luna Network HSM Appliance Software 7.8.0 or newer, Luna HSM Firmware 7.8.0 or newer, and each appliance in the cluster must have the **cluster** secure package installed. Luna HSM Client 10.5.0 or newer is required to access keyrings on a cluster. This feature is currently available on password-authenticated Luna Network HSMs only.

Creating New Keyrings

Use this procedure to create new keyrings on an existing cluster using "REST API" below or "LunaSH" on the next page.

REST API

To create new keyrings

1. Create keyrings on the cluster by specifying a label and initial password, and the label and CO password for the partition that will generate the keyrings. To create multiple keyrings (up to 100 at a time), specify the number to be created (**total** in **--data**). Each keyring will be given the specified label with an appended index number. You can use **index** in **--data** to specify the starting number for the group of keyrings.

   You can currently create up to 3000 keyrings on the cluster.

   POST /api/keyrings

   ```
   curl -X POST "https://1.2.3.4:50070/api/keyrings" --data '{"authentication":[{"type":
   "CO","partitionLabel": "clusterpar","password": "iamtheCO"}],"keyRingLabel":
   "newkeyRing","keyRingPassword": "iamtheCO","index": "100","total": "10"}' -H "Content-Type:
   application/json" -H "Authorization: Basic dXNlcjpwYXNzd29yZA==" --insecure
   ```

2. [Optional] List the keyrings available on the cluster. Each individual keyring is listed along with its UUID.

   GET /api/keyrings

   ```
   curl -X GET "https://1.2.3.4:50070/api/keyrings" -H "Content-Type: application/json" -H
   "Authorization: Basic dXNlcjpwYXNzd29yZA==" --insecure
   ```

3. [Optional] Display the attributes for an individual keyring by specifying its UUID.

   GET /api/keyrings/{keyringid}

```
curl -X GET "https://1.2.3.4:50070/api/keyrings/d533441d-9c3f-403b-9280-6fc6ccaae338" -H
"Content-Type: application/json" -H "Authorization: Basic dXNlcjpwYXNzd29yZA==" --insecure
```

You must now use **ckdemo** on the Luna HSM Client machine to configure the keyring for cryptographic use. See "Configuring Keyring Roles" below.

LunaSH

To create new keyrings

1. Create keyrings on the cluster by specifying its label and initial password, and the label and CO password for the partition that will generate the keyring. To create multiple keyrings (up to 100 at a time), specify the number to be created using the **-total** option. Each keyring will be given the specified label with an appended index number. You can use the **-index** option to specify the starting number for the group of keyrings.

   You can currently create up to 3000 keyrings on the cluster.

   lunash:> **keyring create -label** <label> **-password** <password> **-partition** <partition_label> **-copassword** <password> **-total** <number> **-index** <number>

2. [Optional] List the keyrings available on the cluster. Each individual keyring is listed along with its UUID.

   lunash:> **keyring list**

3. [Optional] Display the attributes for an individual keyring by specifying its UUID.

   lunash:> **keyring show -keyringid** <UUID>

Configuring Keyring Roles

Use this procedure to initialize the keyring roles required for creating and using objects on the keyring. The keyring has its own administrative roles, analogous to the Partition SO and Crypto Officer roles (see Partition Roles). For clarity, they are referred to here as the Keyring Security Officer (KRSO) and Keyring Crypto Officer (KRCO) roles, although they use the same commands as the partition roles. The procedure is intended to enforce role separation between the Keyring Security Officer and the Keyring Crypto Officer, the same way roles are separated on standard Luna HSM partitions.

This procedure uses LunaCM on a registered Luna HSM Client.

To configure roles on the keyring

1. Run LunaCM and confirm that all your created keyrings are available as slots. Note the slot number and label of the keyring you wish to configure.
2. The initial password you set during keyring creation belongs to the KRSO role (called the PO in LunaCM). Log in as KRSO.

   lunacm:> **role login -name po**

3. The KRSO password must now be changed to enforce role separation between the cluster administrator and the keyring user.

   lunacm:> **role changepw -name po**

> **NOTE** While the KRSO and KRCO roles are separate, they are intended to be held by the same individual and they must therefore use the same password. Consider this when setting your KRSO password.
>
> The KRCO password is what your applications will specify to access the keyring and create and use objects. Thales recommends that you always use the most secure password possible. The length of your KRSO/KRCO password affects the behavior of the keyring as follows:
>
> - If the KRCO password is 16 characters or shorter, the keyring is locked after 10 failed login attempts and must be unlocked before it can be used again:
>
>   **PATCH /api/keyrings/{keyringid}**
>
>   lunash:> **keyring unlock -copassword** <password> {**-keyringid** <string> | **-label** <name>}
> - If the KRCO password is 17 characters or longer, the lockout counter is not incremented.
>
> Failed login attempts using 6 characters or less never increment the counter.

4. Initialize the KRCO role on the keyring. Ensure that you set the same password for this role that you set for the KRSO.

   lunacm:> **role init -name co**

   The KRCO can now log in and use the keyring for key creation and most cryptographic operations, just as you would a standard Luna application partition.

Deleting a Keyring

Use this procedure to delete a keyring using "REST API" below or "LunaSH" below.

REST API

To delete a keyring

1. [Optional] List the keyrings available on the cluster. Each individual keyring is listed along with its UUID.

   GET /api/keyrings

   ```
   curl -X GET "https://1.2.3.4:50070/api/keyrings" -H "Content-Type: application/json" -H
   "Authorization: Basic dXNlcjpwYXNzd29yZA==" --insecure
   ```

2. Delete the keyring by specifying either its label or UUID, and the CO password for the cluster partition.

   DELETE /api/keyrings/{keyringid}

   ```
   curl -X DELETE "https://1.2.3.4:50070/api/keyrings/d533441d-9c3f-403b-9280-6fc6ccaae338" -H
   "Content-Type: application/json" -H "Authorization: Basic dXNlcjpwYXNzd29yZA==" --insecure
   ```

LunaSH

To delete a keyring

1. [Optional] List the keyrings available on the cluster. Each individual keyring is listed along with its UUID.

   lunash:> **keyring list**

2. Delete the keyring by specifying either its label or UUID, and the CO password for the cluster partition.

lunash:> **keyring delete** **-copassword** <password> [**-label** <keyringlabel> | **-keyringid** <UUID>]

CHAPTER 7: Cluster Backup and Restore

These procedures allow you to back up the contents of a cluster to an encrypted file on the appliance, using REST API or LunaSH:

> "Backing Up/Restoring the Partition SMK" below

> "Backing Up a Cluster" on the next page

> "Restoring a Cluster from Backup" on page 44

NOTE This feature requires Luna Network HSM Appliance Software 7.8.0 or newer, Luna HSM Firmware 7.8.0 or newer, and each appliance in the cluster must have the **cluster** secure package installed. Luna HSM Client 10.5.0 or newer is required to access keyrings on a cluster. This feature is currently available on password-authenticated Luna Network HSMs only.

Backing Up/Restoring the Partition SMK

All keyrings on the cluster are encrypted with a SKS Master Key (SMK), which is stored on the V1 partition that created the cluster and cloned to the V1 partition on each joining cluster member. In the highly unlikely event that the SMK is lost from all members, the entire cluster is unrecoverable. Therefore, Thales recommends backing up the SMK to a Luna Backup HSM as part of your general recovery plan. This procedure requires an initialized Luna Backup HSM connected to the appliance. You must use LunaSH to back up or restore the cluster SMK.

Prerequisites

> The Luna Backup HSM must be connected to a USB port on the Luna Network HSM appliance, and connected to power if necessary.

To back up the cluster SMK

1. Log in to LunaSH as **admin**, or a custom user with **admin** privileges.
2. [Optional] View the Luna Backup HSMs currently connected to the appliance and find the correct serial number.

   lunash:> **token backup list**
3. If you have not already done so, initialize the backup HSM, specifying its serial number and a label.

   lunash:> **token backup init -label** <backup_hsm_label> **-serial** <backup_hsm_SN>

   You are prompted to set the backup HSM SO password and a cloning domain string.
4. Initiate the backup operation, specifying the cluster partition label, a label for the backup (either a new label or the label, or an existing backup you wish to overwrite), and the Backup HSM serial number.

   If you omit the **-tokenpar** option when creating a new backup, the backup is assigned a default name (<source_partition_name>_<YYYYMMDD>).

lunash:> **partition backup** **-partition** <cluster_partition_label> **-serial** <Backup_HSM_SN> [**-tokenpar** <target_label>]

You are prompted for the following credentials:

a. Crypto Officer password for the cluster partition

b. HSM SO password for the backup HSM (Luna Backup HSM 7 only)

c. New password for the backup partition

d. Domain string for the backup partition (must match the cluster partition domain string)

The backup operation begins once you have completed the authentication process.

To restore the cluster SMK from backup

1. Log in to LunaSH as **admin**, or a custom user with **admin** privileges.
2. [Optional] View the Luna Backup HSMs currently connected to the appliance and find the correct serial number.

   lunash:> **token backup list**
3. [Optional] View the backups currently available on the Backup HSM.

   lunash:> **token backup partition list** **-serial** <Backup_HSM_serialnum>
4. Initiate the restore operation, specifying the cluster partition label, the backup label, the Backup HSM serial number, and the **-add** option.

   lunash:> **partition restore** **-partition** <target_label> **-tokenpar** <backup_label> **-serial** <Backup_HSM_SN> **-add**

   You are prompted for the following credentials:

   a. Crypto Officer password for the backup

   b. Crypto Officer password for the cluster partition

   The restore operation begins once you have completed the authentication process.

Backing Up a Cluster

Use this procedure to back up the contents of a cluster to an encrypted file on the appliance using "REST API" below or "LunaSH" on page 43. You can perform backups on demand or schedule periodic backups.

> **NOTE** Ensure that the member you are using to back up the cluster is authorized at the time of backup; otherwise the backup will fail.

REST API

Prerequisites

> Backup and restore operations can only be done on the primary member of the cluster.

> The **cluster** service must be running on the appliance.

To back up a cluster manually

Back up the cluster by specifying its UUID, a filename for the backup file, and a password that will be required to restore from this backup.

> **NOTE** Using the same filename as an existing backup will overwrite the existing file.

PUT /api/clusters/{clusterid}/backup

```
curl -X PUT "https://1.2.3.4:50070/api/clusters/2afac9b6-f74c-4888-a059-da015240cd1b/backup" --data '{"action": "run", "password": "backuppassword", "filename": "myBackup", "UUID":"2afac9b6-f74c-4888-a059-da015240cd1b"}' -H "Content-Type: application/json" -H "Authorization: Basic dXNlcjpwYXNzd29yZA==" --insecure
```

To schedule periodic backups

1. Set the backup schedule by specifying a filename prefix that will be used to name all backup files (`"filename"`), the number of backups you would like to store (`"total"`), and the time you want backup to occur (`"time"`) in the format **d:hh:mm** (where **d** is a number representing the day of the week as follows: **0** == Sunday, **1** == Monday, **2** == Tuesday, etc).

   > **NOTE** You can schedule a maximum of 15 weekly backups. After the specified number of backups has been saved, you must set a new schedule to continue saving backups. Thales recommends using a new filename prefix to avoid overwriting your old backups, if you want to keep them.

   The filename you specify may have a maximum length of 64 characters. The following characters are allowed:

   `-.0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ_abcdefghijklmnopqrstuvwxyz`

   PUT /api/clusters/{clusterid}/backup

   ```
   curl -X PUT "https://1.2.3.4:50070/api/clusters/2afac9b6-f74c-4888-a059-da015240cd1b/backup" --data '{"action": "config", "total":"5", "time":"3:14:14", "filename": "myBackup"}' -H "Content-Type: application/json" -H "Authorization: Basic dXNlcjpwYXNzd29yZA==" --insecure
   ```

2. Enable the backup service, specifying a password that will be required to restore from any of the scheduled backups.

   PUT /api/clusters/{clusterid}/backup

   ```
   curl -X PUT "https://1.2.3.4:50070/api/clusters/2afac9b6-f74c-4888-a059-da015240cd1b/backup" --data '{"action": "enable", "password": "backuppassword"}' -H "Content-Type: application/json" -H "Authorization: Basic dXNlcjpwYXNzd29yZA==" --insecure
   ```

3. [Optional] Display the scheduled backup configuration.

   PUT /api/clusters/{clusterid}/backup

   ```
   curl -X PUT "https://1.2.3.4:50070/api/clusters/2afac9b6-f74c-4888-a059-da015240cd1b/backup" --data '{"action": "show"} -H "Content-Type: application/json" -H "Authorization: Basic dXNlcjpwYXNzd29yZA==" --insecure
   ```

All backups are available on the appliance in the **admin** user's files. The scheduled backup configuration is not affected by rebooting the appliance or restarting the **cluster** service. However, you must restart the **cluster** service after changing the appliance time zone or after a daylight savings time change, or the new time will not be reflected in the backup schedule.

> **NOTE** If you make further changes to the backup schedule, you must disable and then re-enable the backup service before the change is reflected.
>
> **PUT /api/clusters/{clusterid}/backup**
>
> ```
> curl -X PUT "https://1.2.3.4:50070/api/clusters/2afac9b6-f74c-4888-a059-da015240cd1b/backup" --data '{"action": "disable"}' -H "Content-Type: application/json" -H "Authorization: Basic dXNlcjpwYXNzd29yZA==" --insecure
> ```

LunaSH

Prerequisites

- Backup and restore operations can only be done on the primary member of the cluster.
- The **cluster** service must be running on the appliance.
- You must be logged in to LunaSH as **admin** to do backup/restore operations.

To back up a cluster manually

Back up the cluster by specifying its UUID, a filename for the backup file, and a password that will be required to restore from this backup.

> **NOTE** Using the same filename as an existing backup will overwrite the existing file.

The filename you specify may have a maximum length of 64 characters. The following characters are allowed:
`-.0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ_abcdefghijklmnopqrstuvwxyz`

lunash:> **cluster backup run** **-clusterid** <string> **-filename** <filename> **-backuppassword** <password>

To schedule periodic backups

1. Set the backup schedule by specifying the cluster UUID (**-clusterid**), a filename prefix that will be used to name all backup files (**-filename**), the number of backups you would like to store (**-number**), and the time you want backup to occur (**-time**) in the format **d:hh:mm** (where **d** is a number representing the day of the week as follows: **0** == Sunday, **1** == Monday, **2** == Tuesday, etc).

   > **NOTE** You can schedule a maximum of 15 weekly backups. After the specified number of backups has been saved, you must set a new schedule to continue saving backups. Thales recommends using a new filename prefix to avoid overwriting your old backups, if you want to keep them.

   The filename you specify may have a maximum length of 64 characters. The following characters are allowed:
   `-.0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ_abcdefghijklmnopqrstuvwxyz`

   lunash:> **cluster backup config** **-clusterid** <string> **-filename** <filename> **-number** <number> **-time** <string>
2. Enable the backup service, specifying a password that will be required to restore from any of the scheduled backups.

lunash:> **cluster backup enable -backuppassword** <password>

3. [Optional] Display the scheduled backup configuration.

   lunash:> **cluster backup show**

All backups are available on the appliance in the **admin** user's files. To view them, use lunash:> **my file list**. The scheduled backup configuration is not affected by rebooting the appliance or restarting the **cluster** service. However, you must restart the cluster service after changing the appliance time zone or after a daylight savings time change, or the new time will not be reflected in the backup schedule (see known issue LUNA-25157).

> **NOTE** If you make further changes to the backup schedule, you must disable and then re-enable the backup service before the change is reflected.
>
> lunash:> **cluster backup disable**

Restoring a Cluster from Backup

Use this procedure to restore the primary cluster member from backup using "REST API" below or "LunaSH" on the next page.

> **NOTE** Keyrings can be restored to a different cluster than the one where they were created, as long as the new cluster uses the same SKS Master Key (SMK). If the SMK is destroyed and cannot be restored from backup, keyrings are unrecoverable.

REST API

Prerequisites

- Backup and restore operations can only be done on the primary member of the cluster.
- The **cluster** service must be running on the appliance.

To restore a cluster from backup

1. Restore the cluster from backup by specifying the backup filename, the backup password, and the type of restore operation you want to perform.

   PUT /api/cluster/restore

   ```
   curl -X PUT "https://1.2.3.4:50070/api/cluster/restore" --data '{"myBackupPassword":
   "iamtheCO", "filename": "myBackup.tar.gz", "type": "keyring"}' -H "Content-Type:
   application/json" -H "Authorization: Basic dXNlcjpwYXNzd29yZA==" --insecure
   ```

   > **NOTE** During the restore operation, objects with the same UUID will be skipped.

2. [Optional] Check the status of the restore operation.

   GET /api/cluster/restore

   ```
   curl "https://1.2.3.4:50070/api/cluster/restore" -H "Content-Type: application/json" -H
   "Authorization: Basic dXNlcjpwYXNzd29yZA==" --insecure
   ```


LunaSH

Prerequisites

- Backup and restore operations can only be done on the primary member of the cluster.
- The **cluster** service must be running on the appliance.
- You must be logged in to LunaSH as **admin** to do backup/restore operations.

To restore a cluster from backup

1. Restore the cluster from backup by specifying the backup filename, the backup password, and the type of restore operation you want to perform.

   > **NOTE** Only the **keyring** option is supported; this option restores all keyrings and their contents to the target cluster.

   lunash:> **cluster restore run** **-filename** <filename> **-type keyring -restorepassword** <password>

   > **NOTE** During the restore operation, objects with the same UUID will be skipped.

2. [Optional] Check the status of the restore operation.

   lunash:> **cluster restore show**

