SAP HANA Administration Guide for SAP HANA Platform

Generated on: 2024-11-25 08:54:21 GMT+0000

SAP HANA Platform | 2.0 SPS 08

PUBLIC

Original content: https://help.sap.com/docs/SAP_HANA_PLATFORM/6b94445c94ae495c83a19646e7c3fd56?locale=en-US&state=PRODUCTION&version=2.0.08

Warning

This document has been generated from the SAP Help Portal and is an incomplete version of the official SAP product documentation. The information included in custom documentation may not reflect the arrangement of topics in the SAP Help Portal, and may be missing important aspects and/or correlations to other topics. For this reason, it is not for productive use.

For more information, please visit the https://help.sap.com/docs/disclaimer.

Server-Side Data Encryption Services

SAP HANA features encryption services for encrypting data at rest, as well as an internal encryption service available to applications with data encryption requirements.

Passwords

On the SAP HANA database server, all passwords are stored securely:

- Operating system user passwords are protected by the standard operating system mechanism, `/etc/shadow` file.
- All database user passwords are stored in salted hash form using PBKDF2 (Password-Based Key Derivation Function 2) and, for downward compatibility, secure hash algorithm SHA-256.

  The SAP HANA implementation of PBKDF2 uses the SHA-256 secure hash algorithm and 15,000 iterations.

  > **Note**
  >
  > The hash method SHA-256 can be disabled by setting the parameter `[authentication] password_hash_methods` in the `global.ini` configuration file to `pbkdf2`. The default value is `pbkdf2,sha256`.

- Credentials required by SAP HANA applications for outbound connections are securely stored in a database-internal credential store. This internal credential store is in turn secured using the internal application encryption service.

Data-at-Rest Encryption

To protect data saved to disk from unauthorized access at operating system level, the SAP HANA database supports data encryption in the persistence layer for the following types of data:

- Data volumes
- Redo log volumes
- Data and log backups

Security-Relevant Application Data

An internal encryption service is used to encrypt sensitive application data. This includes credentials required by SAP HANA for outbound connections, private keys of the SAP HANA server stored in the database, and data in secure stores defined by developers of SAP HANA XS applications (classic or advanced) or other applications (through SQL).

Secure Stores

SAP HANA uses either the instance SSFS (secure store in the file system) or the local secure store (LSS), to protect the root keys used for all data-at-rest encryption services and the internal application encryption service. It uses the system PKI SSFS to protect the system-internal root certificates required for secure internal communication.

Related Information

Data and Log Volume Encryption
Backup Encryption
Internal Application Encryption Service
Server-Side Secure Stores
Local Secure Store (LSS)

Server-Side Secure Stores

SAP HANA uses the configured secure store, that is, either the instance SSFS (secure store in the file system) or the default local secure store (LSS), to protect the root keys used for all data-at-rest encryption services and the internal application encryption service. It uses the system PKI SSFS to protect the system-internal root certificates required for secure internal communication.

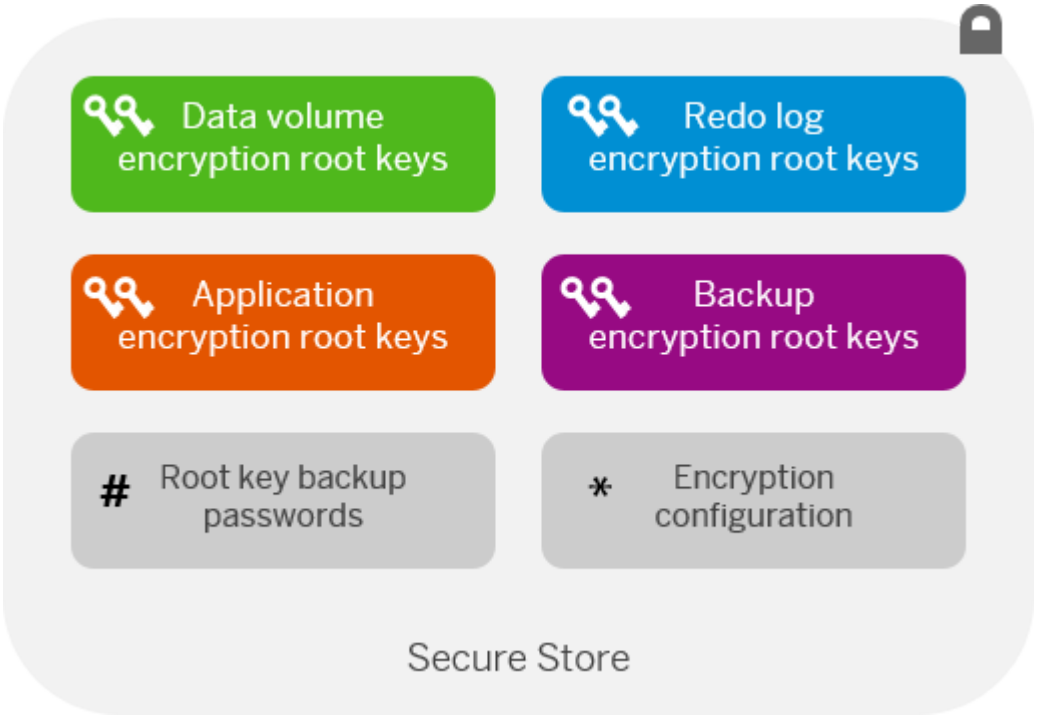
Secure Store for Encryption Root Keys

SAP HANA uses the configured secure store to protect the following:

- The root keys used for:
  - Data volume encryption
  - Redo log encryption
  - Data and log backup encryption
  - Internal application encryption service of the database
- The password of the root key backup
- Encryption configuration information

These root keys protect all encryption keys (and data) used in the SAP HANA database from unauthorized access.

The system database and all tenant databases have their own encryption root keys.



Secure Store Contents

There are two variations of the encryption root key secure store:

- The **instance SSFS** (secure store in the file system) is a single file in the local file system which hosts the encryption keys for all tenants.

ℹ Note

To prevent data encrypted in the SAP HANA database from becoming inaccessible, the content of the instance SSFS and key information in the database must remain consistent. The database detects if this is not case, for example if the instance SSFS becomes corrupted, and issues an alert (check 57). It is recommended that you contact SAP Support to resolve the issue.

- The **local secure store** (LSS) is a separate lightweight utility that runs as a separate service on the SAP HANA server under a different operating system user. It uses tenant-specific files. It is the default secure store. For more information about the LSS, see *Local Secure Store*.

System PKI SSFS

The system PKI SSFS (secure store in the file system) protects the X.509 certificate infrastructure that is used to secure internal SSL/TLS-based communication between hosts in a multiple-host system or between processes of the individual databases in a system. 9.10

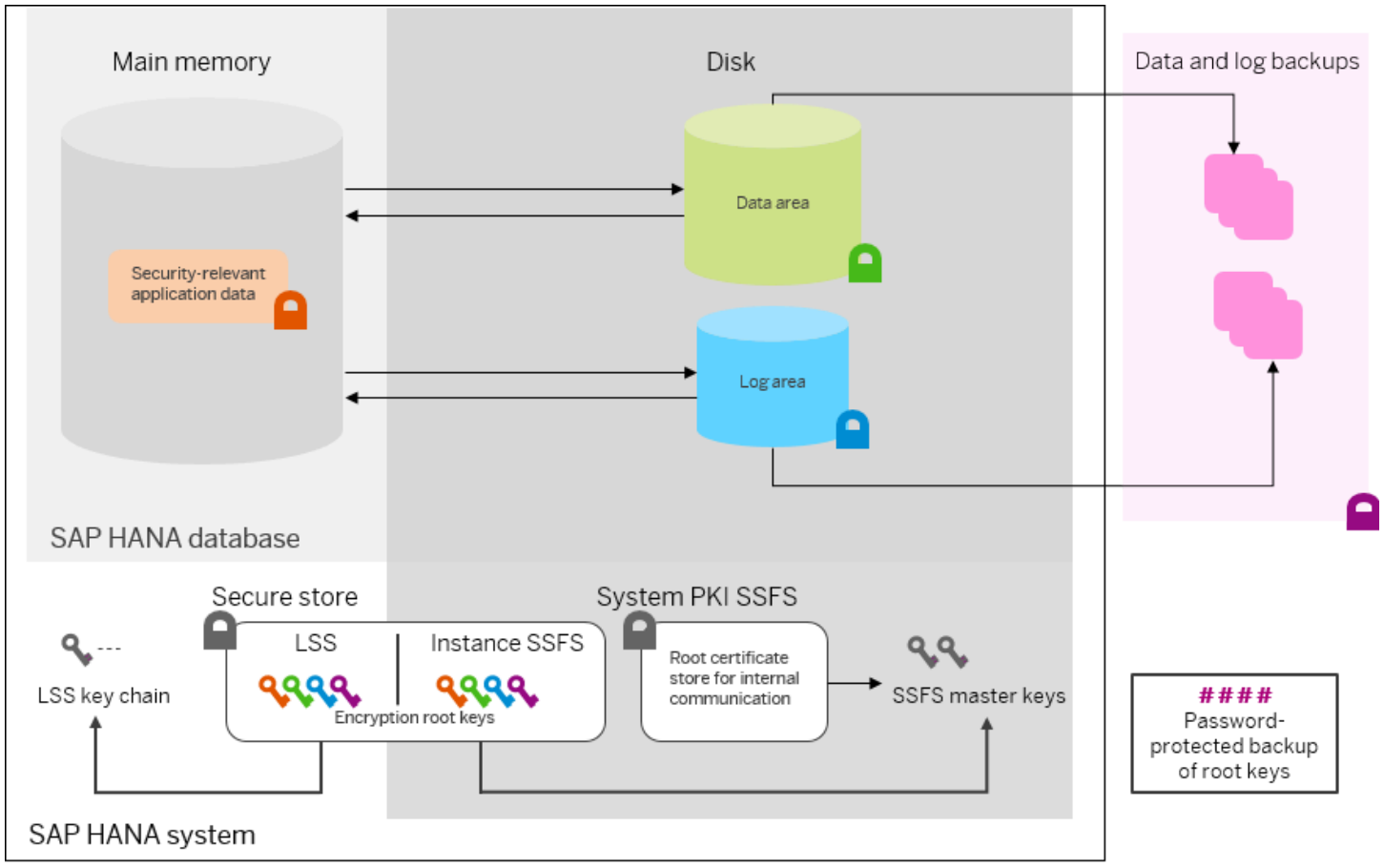
The master key of the system PKI SSFS encrypts all system-internal root certificates required for secure internal communication.

Encryption Services and Keys

The following diagram provides an overview of which data in SAP HANA can be encrypted using a dedicated encryption service, and how all associated encryption root keys are stored in the configured secure store.

ℹ Note

The following diagram shows only one database. However, a system always has a system database and any number of tenant databases. Every database in the system has its own encryption root keys for each of the available encryption services. The root keys of all databases are stored in the configured secure store.



Master Key Change

The contents of both the instance SSFS and the system PKI SSFS are protected by individual master key files in the file system.

Unique master keys are generated for the instance SSFS, if used, and the system PKI SSFS during installation or update. However, if you received your system pre-installed from a hardware or hosting partner, we recommend that you change them immediately after handover to ensure that they are not known outside of your organization. You can also change the master keys any time later.

ℹ Note

The default path of the key file of the instance SSFS is `/usr/sap/<sid>/SYS/global/hdb/security/ssfs`. If you change the default path, you may need to reconfigure it in the event of a system rename.

Related Information

Local Secure Store (LSS)
Secure Internal Communication
Change the SSFS Master Keys

Change the SSFS Master Keys

The secure stores in the file system (SSFS) used by SAP HANA are protected by unique master keys, generated during installation or update. However, if you received your system pre-installed from a hardware or hosting partner, we recommend that you change these master keys immediately after handover to ensure that they are not known outside your organization.

ℹ Note

If you have replaced the instance SSFS with the local secure store (LSS), you do not have an instance SSFS master key, but you do still have a master key for the system PKI SSFS.

Prerequisites

- You have shut down the SAP HANA system.
- You have the credentials of the operating system user (*<sid>*adm) that was created when the system was installed.

Context

You change the SSFS master keys using the command line tool `rsecssfx`, which is installed with SAP HANA and available at `/usr/sap/<SID>/HDB<instance>/exe`.

Before changing the SSFS master keys, note the following:

- In a distributed SAP HANA system, every host must be able to access the file location of the instance SSFS master key.
- The SSFS master keys only have to be changed once for the whole instance and not per tenant database.
- In a system-replication setup you can change the instance SSFS master key on the primary system and the new key will be replicated to the secondary. To trigger replication you must subsequently restart the secondary system. In this case, however, you will need to copy the system PKI SSFS key and data file from the primary system to the secondary manually before registering and restarting the secondary system. Refer to the steps in the topic *Configure SAP HANA System*

Replication with hdbnsutil and SAP Note 2369981 *Required configuration steps for authentication with HANA System Replication*.

→ Remember

In multi-tier system replication scenarios involving three systems, restart the tier-2 secondary system first, then the tier-3 secondary system. If a secondary system takes over from its replication source before the new master key has been replicated, all systems registered will use the old key from the former secondary system instead.

Procedure

1. Log on to the SAP HANA system host as the operating system user, <*sid*>adm.
2. Change the master key of the **instance SSFS**, if used, as follows:
   a. Re-encrypt the instance SSFS with a new key with the command:

   ```
   export RSEC_SSFS_DATAPATH=/usr/sap/<SID>/SYS/global/hdb/security/ssfs
   export RSEC_SSFS_KEYPATH=<path to current key file>
   rsecssfx changekey $(rsecssfx generatekey -getPlainValueToConsole)
   ```

   For script languages bash and csh the syntax is:

   ```
   rsecssfx changekey `rsecssfx generatekey -getPlainValueToConsole`
   ```

   #### ⓘ Note

   The command uses the backtick character not single quotes.

   b. Configure the specified key file location in the `global.ini` configuration file at `/usr/sap/<SID>/SYS/global/hdb/custom/config/global.ini`.

   If the file does not exist, create it. Add the following lines:

   ```
   [cryptography]
   ssfs_key_file_path = <path to key file>
   ```

   #### ⓘ Note

   The default path of the key file is `/usr/sap/<sid>/SYS/global/hdb/security/ssfs`. If you change the default path, you may need to reconfigure it in the event of a system rename.

3. Re-encrypt the **system PKI SSFS** with a new key with the following command:

   ```
   export RSEC_SSFS_DATAPATH=/usr/sap/<SID>/SYS/global/security/rsecssfs/data
   export RSEC_SSFS_KEYPATH=<path to current key file>
   rsecssfx changekey $(rsecssfx generatekey -getPlainValueToConsole)
   ```

   #### ⓘ Note

   The default path of the key file is `/usr/sap/<sid>/SYS/global/security/rsecssfs/key`. If you change the default path, you may need to reconfigure it in the event of a system rename.

   For script languages bash and csh the syntax is:

   ```
   rsecssfx changekey `rsecssfx generatekey -getPlainValueToConsole`
   ```

   #### ⓘ Note

   The command uses the backtick character not single quotes.

4. Restart the SAP HANA system.

Next Steps

Additional Information for Backup and Recovery

For file system-based copies of SAP HANA database installations, you must manually save and restore the instance SSFS master key file. Otherwise data loss can occur.

In regular backup and recovery scenarios, the SSFS must always be restored from a recent root key backup before a database recovery, unless:

- You have never changed the redo log encryption key.
- You are performing a recovery into the same database from which the backup was taken, and the database's secure store is intact and contains the latest root key changes.

> **Note**
> It is not necessary to save the system PKI SSFS key file. The system will generate a new system PKI SSFS automatically if required.

Related Information

Starting and Stopping SAP HANA Systems
Configuring SAP HANA System Properties (INI Files)
Server-Side Data Encryption Services
Import Backed-Up Root Keys Before Database Recovery
Configure SAP HANA System Replication with hdbnsutil
SAP Note 2183624
SAP Note 2369981

Using the LSS with an External Key Management System

The LSS payload database (DB) contains the SAP HANA encryption root keys used for encrypting SAP HANA data, log volumes, and backups.

The body of the LSS payload DB is encrypted with a AES-256-bit key. With every save-operation, a new key for this encryption is generated and stored in a header of the same LSS payload DB. This header is encrypted. By default, the header encryption key is done with a local key chain of LSS (customer-supplied encryption keys [CSEK] scenario).

If desired, the SAP HANA/LSS can be reconfigured to connect LSS to a key in Data Custodian (DC) KMS. If this is done, then an RSA key pair managed by DC KMS is used as the header encryption key. LSS can then only open its payload DB by sending the encrypted header for decryption to DC KMS. This offers the capability to disrupt the key chain by disabling or deleting the key in KMS (customer-controlled encryption keys [CCEK] scenario). However, this makes the SAP HANA/LSS strictly dependent on the key in KMS.

> **Note**
> Only the SAP Data Custodian Key Management Service is supported.

When SAP DC KMS is used with the LSS, the LSS payload DB is encrypted with the public key of an asymmetric key pair whose private key is stored in DC KMS. Consequently, to read and decrypt a payload DB, access to the private key of the key pair is

required.

DC KMS unreachable

If DC KMS is unreachable for some reason (for example, network issues, or a downtime of KMS), SAP HANA will in most cases continue to work, thanks to the filled caches in LSS.

Only if the LSS cache is empty (usually after a full restart of SAP HANA and LSS), the SAP HANA database can hang or even crash if KMS is not reachable. The detailed error traces of LSS should help to analyze the situation.

To verify that DC KMS is available and that there is an active key, SAP HANA regularly polls into KMS. This allows LSS to respond to the following changes:

- **Key revocation**

  If the DC KMS reports that the key has been revoked (disabled or deleted), the database is shut down and cannot be restarted or recovered from a backup (until the key is available again).

  > ⚠ **Caution**
  > Key revocation is not trivial and should never be used to temporarily prevent access to applications. While a disabled key can be re-enabled, the cost (in terms of effort and time) of resuming application operation after a temporary key disablement may be high. Deleting a key is irreversible.

- **Key rotation**

  If DC KMS reports that the key has been rotated (for example, a new primary version of the key has been created), the currently active key management configuration is updated and LSS automatically re-encrypts the payload DB with the new key. This can be confirmed by querying the `KEY_MANAGEMENT_CONFIGURATIONS` system view: the active configuration has the new key version configured.

You can configure the polling interval using the `[cryptography] key_revocation_check_interval` in the `nameserver.ini`. The value must be between 1 and 3600 seconds. The default value is 60 seconds. If you set a value outside this range, it defaults to 60 seconds. Furthermore, if you set a value under 60 seconds, a warning is displayed that the polling rate might be too high and could lead to unnecessarily high network traffic.

Second Access Key Pair

The LSS-internal caching makes HANA/LSS partly resilient against temporary KMS outages. However, a permanent loss of the key in KMS, e.g., due to a disaster in KMS, would lead to a complete loss of the HANA database.

To also eliminate this risk, you can configure a second access key pair. You generate this second access key pair yourself.

> ℹ **Note**
> Keep the second access key pair in a safe place in case of an emergency.

Only the public key of the second access key pair is provided to LSS, as a self-signed certificate within the external key management configuration passed to SAP HANA. SAP HANA forwards this certificate to the LSS, which then uses it to add a second header to the payload DB. This allows the Payload DB to be decrypted either regularly with the help of DC KMS (using the first header), or, in an emergency case, with the private key of the second access key pair (using the second header).

To bring SAP HANA/LSS back into an operable state after a DC KMS emergency, you need to take special action in which you provide the private key of the second access key pair to LSS, by:

- Restoring the LSS component using the *lssbackup* utility (*restoreWithSecondAccessKey*)

- Recovering an *lssbackup* as the first step in recovering an encrypted SAP HANA database (`RECOVER ENCRYPTION ROOT KEYS AND SETTINGS … SECURE STORE SECOND ACCESS`)

Related Information

Create and Manage a Key Management Configuration
Create a Second Access Key
Add Second Access Certificates to the Key Management Configuration
Monitoring Key Management Configurations
Restore a Payload Database File Backup with the Second Access Key
LSS Backup Utility (lssbackup)
Import Backed-Up Root Keys Before Database Recovery
RECOVER ENCRYPTION ROOT KEYS Statement (Backup and Recovery)
LSS Trace and Dump Files
SAP Note 2917358
SAP Note 2911896

Create and Manage a Key Management Configuration

A key management configuration includes a JSON document containing the configuration information required to connect to the chosen external key management service (KMS) or hardware security module (HSM).

Prerequisites

You can log on to the system database and have the system privileges `ENCRYPTION ROOT KEY ADMIN` and `DATABASE ADMIN`.

Context

You create and manage the tenant-specific configurations required to use a KMS or HSM in the system database using the SQL statements described below. Every database has a `DEFAULT` configuration, which is active by default and corresponds to no connection to an external KMS.

⚠ Caution

After a key management configuration has been added, changed or activated, the LSS must be backed up. If automatic LSS backups are enabled (default), the required backup is always written and available for recovery. If automatic LSS backups are disabled (not recommended), this must be considered in your backup strategy. A missing backup could result in your database being unrecoverable. For more information, see *Automatic Content Backups* in the *SAP HANA Security Guide*.

For more information about the settings required by a specific KMS or HSM, see SAP Note 2917358.

Procedure

- To **add a key management configuration** for a specific database (tenant database or system database), execute the following statement:

```
ALTER DATABASE <database_name>
    ADD KEY MANAGEMENT CONFIGURATION <config_name> PROPERTIES '<settings>'
```

Parameter	Description
<database_name>	Name of the database
<config_name>	A unique name consisting of uppercase letters only, for example, 'AWS_HSM'. The name 'DEFAULT' is forbidden.
<settings>	A JSON document with key-value settings. The list of keys and their supported values depends on the chosen KMS or HSM. The command will fail if the specified settings do not work.

The new configuration for the specified database is stored in the LSS tenant-specific configuration database file (`lssconfig.db`).

- To **alter a key management configuration** for a specific tenant database, execute the following statement:

```
ALTER DATABASE <database_name>
    ALTER KEY MANAGEMENT CONFIGURATION <config_name> PROPERTIES '<updates>'
```

Parameter	Description
<database_name>	Name of the tenant database
<config_name>	The name of the configuration to be modified, in uppercase letters
<updates>	A JSON document with the new key-value settings. The list of keys and their supported values depends on the chosen HSM or KMS. All settings not described in the update remain unchanged.

This command will fail if the changed settings do not work.

- To **activate a key management configuration** for a specific tenant database, execute the following statement.

  By activating a configuration, you can switch between different configurations.

```
ALTER DATABASE <database_name>
    ACTIVATE KEY MANAGEMENT CONFIGURATION <config_name>
```

Parameter	Description
<database_name>	Name of the database
<config_name>	The name of the configuration to be activated, in uppercase letters

The command will fail (leaving everything unchanged) if the newly activated configuration does not work.

- To **remove a key management configuration** for a specific tenant database, execute the following statement:

```
ALTER DATABASE <database_name>
    DROP KEY MANAGEMENT CONFIGURATION <config_name>
```

Parameter	Description
<database_name>	Name of the database

Parameter	Description
<config_name>	The name of the configuration to be deleted, in uppercase letters

The command will fail if the specified configuration is still active. You deactivate a configuration by activating another one.

> **Note**
>
> If you no longer want to use the LSS with a key management system at all, activate the `DEFAULT` key management configuration.

Related Information

SAP Note 2917358
ALTER SYSTEM {ADD | ACTIVATE | UPDATE | DROP} KEY MANAGEMENT CONFIGURATION Statement (System Management)
ALTER DATABASE Statement (Tenant Database Management)

Create a Second Access Key

The second access key is needed for recovering from disasters in the key management service (KMS) or hardware security module (HSM). You create a second access key by using the SAPGENPSE tool.

Procedure

> **Note**
>
> For security reasons, we recommend that you do not enter the passwords (`-x` and `-z`) on the command line. By omitting these parameters, you will be interactively prompted to enter the passwords. In this way, you avoid unintentionally leaving the passwords in the command history.

1. Create an asymmetric key pair as follows:

   ```
   sapgenpse get_pse -a RSA:4096:SHA256 -x <second_access_key_passphrase> -p <second_access_key>
   ```

Parameter	Description
<second_access_key_passphrase>	Password for the PSE file
<second_access_key>	Name of the PSE file

2. Export the certificate with the public key to the file *<second_access_certificate>*`.pem`:

   ```
   sapgenpse export_own_cert -p <second_access_key>.pse -x <second_access_key_passphrase> -o <se
   ```

Parameter	Description
<second_access_key>	Name of the PSE file
<second_access_key_passphrase>	Password for the PSE file

Parameter	Description
<second_access_certificate>	File name of the exported certificate

3. Export the private key file as follows and then store it in a safe place.

   You can optionally add a passphrase for the exported private key interactively or by using the `-z` option. If you add a passphrase, it must be inserted whenever the private key is used. Make sure that you do not forget the passphrase, or store it in a safe place.

   ```
   sapgenpse export_p8 -x <second_access_key_passphrase> -z <private_key_passphrase> -p <second_
   ```

Parameter	Description
<second_access_key_passphrase>	Password for the PSE file
<second_access_key>	Name of the PSE file
<second_access_private_key>	File name of the exported private key
<private_key_passphrase>	Password for the exported private key file

   Once the private key and passphrase have been stored in a safe place, the PSE file can be deleted.

Related Information

Encryption Key Chain

Add Second Access Certificates to the Key Management Configuration

The public key of the second access key is provided as a self-signed certificate that is part of the external key management configuration passed to SAP HANA.

Prerequisites

You can log on to the system database and have the system privileges `ENCRYPTION ROOT KEY ADMIN` and `DATABASE ADMIN`.

Procedure

1. In the JSON document with key-value settings, enhance an already existing external key management configuration by adding the second access certificate to a JSON object with the `SET` action. For example:

   ```
   {
       "setup": {
           "configure_second_access": {
               "action": "SET",
               "certificates": [
                  "-----BEGIN CERTIFICATE-----
   <certificate content base64 encoded>
   -----END CERTIFICATE-----"
               ]
           }
       }
       <KMS-specific configuration goes here>
   }
   ```

- If you have already configured second access certificates for the active external key management configuration and want to add them to a new key management configuration, apply the KEEP action in the JSON document with key-value settings:

```
{
    "setup": {
        "configure_second_access": {
            "action": "KEEP"
        }
    }
    <KMS-specific configuration goes here>
}
```

- If you already have second access certificates for the active external key management configuration but do not want to use them in a new key management configuration, apply the NONE action in the JSON document with key-value settings:

```
{
    "setup": {
        "configure_second_access": {
            "action": "NONE"
        }
    }
    <KMS-specific configuration goes here>
}
```

2. Add, alter, or activate the key management configuration using the corresponding SQL statements. See *Create a Key Management Configuration*.

Results

The LSS now encrypts the body encryption key of the payload DB file twice: first with the certificate obtained from the KMS or HSM, and second with the certificate of the second access key pair. Both results are stored in the header of the payload database file. In normal operations, the LSS opens the payload DB file with the help of the KMS or HSM. Use the second access key only to recover from disasters in the KMS or HSM.

Related Information

Create and Manage a Key Management Configuration

Monitoring Key Management Configurations

The monitoring view KEY_MANAGEMENT_CONFIGURATIONS provides information about the existing key management configurations.

It gives the following details:

Column	Description
DATABASE_NAME	Database name
CONFIGURATION_NAME	Name of the key management configuration
IS_ACTIVE	Indicates whether this configuration is used by the LSS to define access to SAP HANA's sensitive information
TYPE	Type of external key management
CLIENT_VERSION	Version of the driver software

Column	Description
PROPERTIES	Reduced version of the `properties` section of the *<settings>* JSON (the "credentials" object is omitted)
CONFIGURATION_ID	ID that uniquely identifies a key management configuration

The certificates configured for the second access key are shown as an additional object, "second_access", of the *<settings>* JSON, as shown below:

```
{
    "second_access": {
        "certificates": [
            "-----BEGIN CERTIFICATE-----
<certificate content base64 encoded>
-----END CERTIFICATE-----"
        ]
    }
    <KMS-specific configuration goes here>
}
```

Related Information

KEY_MANAGEMENT_CONFIGURATIONS System View

Restore a Payload Database File Backup with the Second Access Key

Use the private key file of the second access key to restore the payload database (DB) file of a tenant in an emergency. This is necessary in rare cases only.

Prerequisites

- You have operating system-level access to the SAP HANA system as the *<sid>*`crypt` user.
- When the LSS backup was created, the active key management configuration contained a second access key. See *Create a Second Access Key* and *Add Second Access Certificates to the Key Management Configuration*.

Procedure

1. Copy the private key file from its safe place to a directory that is readable and writable by *<sid>*`crypt`.
2. Restore the tenant's payload DB file using the *lssbackup* utility:

   ```
   lssbackup -t <tenant> restoreWithSecondAccessKey <backupfile> payload <configbackup> <private
   ```

Parameter	Description
<tenant>	Name of the tenant **Note**

Parameter	Description
	The system database and tenant databases are both tenants in the LSS context.
<backupfile>	Backup file of the payload DB file
<configbackup>	Backup file of the `config` DB file
<privatekeyfile>	The file name of the PEM-encoded private key file of the second access key that was used in the LSS backup

> **Note**
>
> The payload DB file cannot be restored from a single backup file that contains both the contents of the payload DB file and the config DB file.

3. Enter the LSS backup password.
4. Enter the passphrase for the private key if one has been set.

Results

The restore operation uses the second access key provided as a private key file and the corresponding passphrase to decrypt and restore the contents of the payload DB file.

Related Information

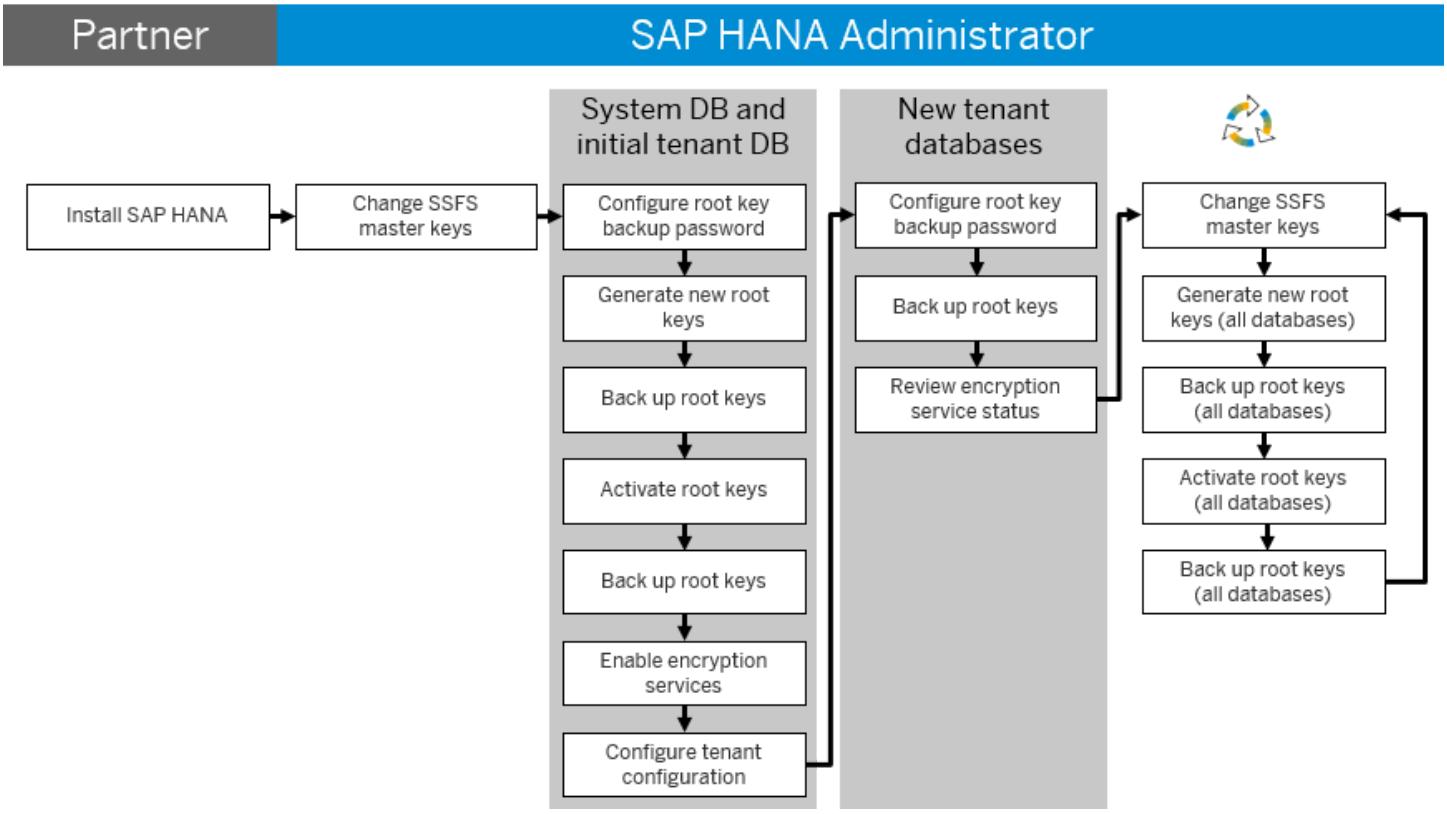
LSS Backup Utility (lssbackup)

Encryption Configuration

We recommend that you configure data encryption immediately after handover of your system from a hardware or hosting partner.

First-Time Configuration

The following figure shows the recommended process for configuring encryption in your SAP HANA system for the first time.



Immediately after system handover from your hardware or hosting partner, perform the following high-level steps.

Location	Steps	More Information
On the SAP HANA server	Change the master keys of the instance SSFS, if used, and the system PKI SSFS.(1)	Unique master keys are generated for the instance SSFS, if used, and the system PKI SSFS during installation or update. However, if you received your system pre-installed from a hardware or hosting partner, we recommend that you change them immediately after handover to ensure that they are not known outside of your organization. You can also change the master keys any time later.
In the system database	1. Set the password for the root key backup for the system database. This password is required to securely back up root keys and subsequently restore backed-up root keys during data recovery.(2)	⚠ **Caution** The password is stored in the secure store along with the other root keys and used whenever you create a backup of the encryption root keys. The password should also be stored in a separate safe location. You will need to enter it to restore the secure store content before a database recovery. Losing this password may result in the database being unrecoverable.
	2. Change the encryption root keys for all encryption services (data volume encryption, redo log encryption, data and log backup encryption, and internal application encryption) in the system database.(3)	Unique root keys are generated during installation or database creation. However, if you received SAP HANA from a hardware or hosting partner, we recommend that you change them immediately after handover to ensure that they are not known outside of

Location	Steps	More Information
	1. Generate new root keys. 2. Back up all root keys to a root key backup file in a secure location. 3. Activate the new root keys. 4. Back up all root keys.	your organization. You can also change root keys any time later. You must back up all keys after you generate or activate a key of any type. This ensures that you always have an up-to-date backup of your root keys available for recovery. **⚠ Caution** Store the root key backup file in a safe location. Losing this file may result in the database being unrecoverable.
	3. Enable the required encryption services: Data volume encryption, redo log encryption, and data and log backup encryption in the system database.(4)	Although SAP HANA provides you with the flexibility to encrypt data volumes, redo logs, and backups independently of each other, if you require full protection in the persistence layer, we recommend that you enable all services. It is not necessary to enable the internal application encryption service explicitly. It is available automatically to requesting applications.
	4. Configure how you want encryption to be handled in new tenant databases. You can do so with the following parameters in the `database_initial_encryption` section of the `global.ini` configuration file: • `persistence_encryption` (default: off) • `log_encryption` (default: off) • `backup_encryption` (default: off) • `encryption_config_control` (default: local_database)	By default, all encryption services are disabled in new tenant databases and only tenant database administrators can enable them. See *Encryption Configuration Control*.
In the first tenant database (if automatically created during installation)	1. Set the password for the root key backup for the first tenant database.(2)	The password is required to securely back up root keys and subsequently restore backed-up root keys during data recovery.
	2. Change the encryption root keys for all encryption services in the first tenant database.(3)	See step 2 above (system database).
	3. Enable the required encryption services in the first tenant database.(4)	By default, only the tenant database administrator can do this in the tenant database. See *Encryption Configuration Control*.

Location	Steps	More Information
In subsequent tenant databases	1, Set the password for the root key backup for the tenant database.[2]	The password is required to securely back up root keys and subsequently restore backed-up root keys during data recovery.
	2. Back up all root keys to a root key backup file in a secure location.	It is not necessary to change the root keys in new tenant databases. Unique root keys are generated on database creation and cannot be known outside of your organization. **⚠ Caution** Store the root key backup file in a safe location. Losing this file may result in the database being unrecoverable.
	3. Change the status of encryption services in the tenant database if required.[4] Encryption services are configured in line with the parameters in the `database_initial_encryption` section of the `global.ini` configuration file as described above.	Who can enable or disable encryption services depends on how the parameter `encryption_config_control` is configured. See *Encryption Configuration Control*.
During operation	• Periodically change the master keys of the instance SSFS, if used, and the system PKI SSFS in line with your security policy. • Periodically change the encryption root keys in all databases in line with your security policy.	

[1] If the instance SSFS is used in a system-replication configuration, you change the instance SSFS master key on the primary system. To trigger replication of the new key to the secondary system, you must subsequently restart the secondary system. In multi-tier system replication scenarios involving three systems, restart the tier-2 secondary system first, then the tier-3 secondary system. If a secondary system takes over from its replication source before the new master key has been replicated, all systems registered will use the old key from the former secondary system instead.

[2] In a system-replication configuration, set the root key backup password in the primary system only. The password will be propagated to all secondary systems. The secondary systems must be running and replicating.

[3] In a system-replication configuration, change root keys in the primary system only. New keys will be propagated to all secondary systems. The secondary systems must be running and replicating.

[4] In a system-replication configuration, enable (or disable) encryption in the primary system only. The setting will be propagated to all secondary systems. The secondary systems must be running and replicating.

Configuration After Update from a Single-Container System

If you updated from a single-container system, your system has a system database and one tenant database. The existing data encryption configuration is retained. Note the following:

- The SSFS master keys for the system remain unchanged and the instance SSFS is still used.
- Existing encryption root keys are the encryption root keys of the tenant database. The update process generates new unique root keys for the system database.
- If a root key backup password existed before update, it is the root key backup password of the tenant database. The system database will not have a root key backup password set. You must set this password manually after the update.

- Encryption services that were enabled before update are enabled in both the system database and the tenant database.

Related Information

Change the SSFS Master Keys
Set the Root Key Backup Password
Changing Encryption Root Keys
Enable Encryption
Encryption Configuration Control
Encryption Key Management

Encryption Configuration Control

You can enable or disable the encryption of data and log volumes, and data and log backups in a new SAP HANA database or in an existing operational database. For a tenant database, you need to know whether encryption configuration is controlled by the tenant database or the system database.

Ownership of Encryption Control

By default, encryption configuration is controlled by the tenant database, but the control can be switched to the system database, or the system database can switch control back to the tenant database.

To see which database is controlling encryption configuration for a tenant database, you can query the system view SYS.M_ENCRYPTION_OVERVIEW. From the system database, you can query the system view SYS_DATABASES.M_ENCRYPTION_OVERVIEW.

Encryption Control in New Tenant Databases

When a new tenant database is created, the `encryption_config_control` parameter in the `database_initial_encryption` section of the `global.ini` configuration file in the system database determines whether encryption configuration is controlled by the tenant database or the system database. You can use this parameter to configure encryption control for new tenant databases:

- If the value of this parameter is `local_database` (default), then only the tenant database administrator can enable or disable encryption from the tenant database.
- If the value is `system_database`, then only the system database administrator can enable or disable encryption from the system database.

Switching Encryption Control in Existing Tenant Databases

If the tenant database controls encryption configuration, the tenant database administrator can hand over this control to the system administrator by executing the following ALTER SYSTEM statement:

```
ALTER SYSTEM ENCRYPTION CONFIGURATION CONTROLLED BY SYSTEM DATABASE
```

If the system database controls encryption configuration, the system database administrator can hand it over to the tenant database administrator by executing the following ALTER DATABASE statement:

```
ALTER DATABASE <database_name> ENCRYPTION CONFIGURATION CONTROLLED BY LOCAL DATABASE
```

For simplicity, the system database administrator can hand over control to all tenants at once by executing the following statement:

```
ALTER SYSTEM ENCRYPTION CONFIGURATION CONTROLLED BY LOCAL DATABASES
```

Related Information

ALTER SYSTEM ENCRYPTION CONFIGURATION Statement (System Management)
ALTER DATABASE Statement (Tenant Database Management)
M_ENCRYPTION_OVERVIEW System View

Set the Root Key Backup Password

The root key backup password is required to securely back up the root keys of the database and subsequently to restore the backed-up root keys during data recovery.

Prerequisites

You have the system privilege `ENCRYPTION ROOT KEY ADMIN`.

Procedure

Set the root key backup password with the following SQL statement.

```
ALTER SYSTEM SET ENCRYPTION ROOT KEYS BACKUP PASSWORD <passphrase>
```

The length and layout of the password must be in line with the database's password policy.

⚠ Caution

If the root key backup already has a password, it will be overwritten.

i Note

In a system-replication configuration, set the root key backup password in the primary system only. The password will be propagated to all secondary systems. The secondary systems must be running and replicating.

Results

The password is set and stored in the secure store together with the SAP HANA encryption root keys and encryption-related configuration. You must provide this password to import and export root keys from the backup into the database before starting a database recovery. All root key backups taken after the password is set use this password to protect the backup files.

i Note

If you are using the local secure store (LSS), the password set is used to fully backup and recover the SAP HANA database, including the secure store.

For more information about root key backups, see the *SAP HANA Security Guide*. For more information about setting the root key backup password using the SAP HAN cockpit, see the SAP HANA cockpit documentation.

⚠ Caution

The password should also be stored in a separate safe location. You will need to enter it to restore the secure store content before a database recovery. Losing this password may result in the database being unrecoverable.

→ Tip

To verify that the password you have is the same as the one that the system uses when creating new root key backups, use the statement `ALTER SYSTEM VALIDATE ENCRYPTION ROOT KEYS BACKUP PASSWORD <passphrase>`.

Related Information

Password Policy Configuration Options
Root Key Backup
ALTER SYSTEM VALIDATE ENCRYPTION ROOT KEYS BACKUP PASSWORD Statement (System Management)
Set the Root Key Backup Password

Changing Encryption Root Keys

Unique root keys are generated during installation or database creation. However, if you received SAP HANA from a hardware or hosting partner, we recommend that you change them immediately after handover to ensure that they are not known outside of your organization. You can also change root keys any time later.

Change the root keys for the following encryption services immediately after handover of your system and periodically during operation:

- Data volume encryption
- Redo log encryption
- Data and log backup encryption
- Internal application encryption

It is important to always change encryption root keys as follows:

1. Generate new root keys.
2. After root keys generation, back them up.
3. Activate new root keys.
4. After activation, back up all root keys again.

⚠ Caution

You must back up all keys after you generate or activate a key of any type. This ensures that you always have an up-to-date backup of your root keys available for recovery.

i Note

In a system-replication configuration, change root keys in the primary system only. New keys will be propagated to all secondary systems. The secondary systems must be running and replicating.

Generate New Root Keys

The first step in changing encryption root keys is to generate new root keys.

Prerequisites

- You are connected to the tenant database requiring the root key change.
- You have the system privilege `ENCRYPTION ROOT KEY ADMIN`.

Procedure

Generate new root keys for all encryption services using the following SQL statements:

Encryption Service	Statement
Data volume encryption	`ALTER SYSTEM PERSISTENCE ENCRYPTION CREATE NEW ROOT KEY WITHOUT ACTIVATE`
Redo log encryption	`ALTER SYSTEM LOG ENCRYPTION CREATE NEW ROOT KEY WITHOUT ACTIVATE`
Data and log backup encryption	`ALTER SYSTEM BACKUP ENCRYPTION CREATE NEW ROOT KEY WITHOUT ACTIVATE`
Internal application encryption	`ALTER SYSTEM APPLICATION ENCRYPTION CREATE NEW ROOT KEY WITHOUT ACTIVATE`

New root keys for all encryption services are generated. To verify the creation of new root keys, query the system view `ENCRYPTION_ROOT_KEYS` and check the status of the latest entries for each root key type.

Related Information

ENCRYPTION_ROOT_KEYS System View
ALTER SYSTEM PERSISTENCE ENCRYPTION Statement (System Management)
ALTER SYSTEM LOG ENCRYPTION Statement (System Management)
ALTER SYSTEM BACKUP ENCRYPTION Statement (System Management)
ALTER SYSTEM APPLICATION ENCRYPTION Statement (System Management)

Back Up Root Keys

After you have generated or activated new encryption root keys or created a new tenant database with new root keys, you must back up all root keys.

Prerequisites

- The external location to which you plan to save the backup is accessible.
- You have set the root key backup password.
- If using the SQL statement `BACKUP ENCRYPTION ROOT KEYS`:
    - The system database has encryption configuration control. See *Encryption Configuration Control*.
    - You have the system privilege `ENCRYPTION ROOT KEY ADMIN`

    - You have the system privilege `BACKUP ADMIN` or `BACKUP OPERATOR` if backing up the system database root keys, or the system privilege `DATABASE BACKUP ADMIN` or `DATABASE BACKUP OPERATOR` if backing up a tenant database's root keys.

- If using the SQL function `ENCRYPTION_ROOT_KEYS_EXTRACT_KEYS` or `ENCRYPTION_ROOT_KEYS_EXTRACT_ALL_KEYS_FOR_DATABASE`, you have the system privilege `ENCRYPTION ROOT KEY ADMIN`.
- If using `hdbnsutil`, you have the credentials of the operating system user (*<sid>*adm).
- If using `hdbnsutil`, you know the ID of the database whose root keys you want to back up. You can determine the IDs of all tenant databases by executing the following SQL command in the system database:

```
SELECT DATABASE_NAME,
        CASE WHEN (DBID = '' AND
        DATABASE_NAME = 'SYSTEMDB')
        THEN 1
        WHEN (DBID = '' AND
        DATABASE_NAME <> 'SYSTEMDB')
        THEN 3
        ELSE TO_INT(DBID)
        END DATABASE_ID
FROM (SELECT DISTINCT DATABASE_NAME, SUBSTR_AFTER (SUBPATH,'.') AS DBID FROM SYS_DATABASES.M_
```

Context

A backup of encryption root keys must be available at an external location to ensure recovery is possible in certain scenarios.

> ⚠ **Caution**
> Store both the root key backup and the password required to read it in a secure location. Losing the backup or the password may result in the database being unrecoverable.

You can back up root keys in a number of ways.

Option	Execution Location	Database Mode	Backup File Location
SQL statement `BACKUP ENCRYPTION ROOT KEYS`	System database	Tenant database can be online or offline; system database must be online	File system on the database server. The file needs to be manually copied to a secure external location.
SQL extraction function `ENCRYPTION_ROOT_KEYS_EXTRACT_KEYS`(1)	Applicable database	Database must be online	CLOB result. The result needs to be manually copied and saved to a file at a secure external location.
SQL extraction function `ENCRYPTION_ROOT_KEYS_EXTRACT_ALL_KEYS_FOR_DATABASE`	Applicable database or system database	If executed from the tenant database:	CLOB result. The result needs to be manually copied and saved to a file

Option	Execution Location	Database Mode	Backup File Location
		Database must be online If executed from the system database: Database can be online or offline; system database must be online	at a secure external location.
SAP HANA cockpit[2]	Applicable database	Database must be online	Local file system
***hdbnsutil* tool**[3]	SAP HANA server	Database can be online or offline	File system on the database server. The file needs to be manually copied to a secure external location.

Note

- [1]If the local secure store (LSS) is being used in conjunction with an external key management system, then you cannot back up the root keys using `ENCRYPTION_ROOT_KEYS_EXTRACT_KEYS` since it does extract the keys in the required format. You must use `ENCRYPTION_ROOT_KEYS_EXTRACT_ALL_KEYS_FOR_DATABASE`.
- [2]The SAP HANA cockpit can be used to back up root keys for both the local secure store (LSS) and the secure store in the file system (SSFS).
- [3]If the local secure store (LSS) is being used in conjunction with an external key management system, then you cannot back up the root keys using *hdbnsutil* since a separation of duties between operating system administrator and key administrator is not possible.

Procedure

1. Back up the root keys of a tenant database using one of the following methods:

Option	Description

SQL statement **BACKUP ENCRYPTION ROOT KEYS**	In the system database, execute the SQL statement:
	
	`BACKUP ENCRYPTION ROOT KEYS [ <root_keytype_`
	`USING <root_key_backup_definition_file>`
	
	• *<database_name>* is the name of the tenant datab
	• The *<root_keytype_list>* option lists the root key ty `APPLICATION`. If you do not specify any value for
	
	i Note
	If the LSS is being used in conjunction with an ext key types. A full LSS backup is required. Therefore
	
	• *<root_key_backup_definition_file>* specifies the ba elements: <prefix>_<database_name>_rootkeys (f ENCRYPTION ROOT KEYS Statement. The file is wr

SQL extraction function **ENCRYPTION_ROOT_KEYS_EXTRACT_KEYS**	a. In the tenant database whose keys are being extrac `SELECT ENCRYPTION_ROOT_KEYS_EXTRACT_KE` The result of this command is a character large obj b. Copy the CLOB result and save it to a file at a secur
SQL extraction function **ENCRYPTION_ROOT_KEYS_EXTRACT_ALL_KEYS_FOR_DATABASE**	a. Depending on which database has encryption conf database or the system database: `SELECT ENCRYPTION_ROOT_KEYS_EXTRACT_AL` The result of this command is a character large obj b. Copy the CLOB result and save it to a file at a secur
SAP HANA cockpit	You can back up root keys using the **Data Encryption** app c cockpit documentation.
hdbnsutil tool	**i Note** If the LSS is being used in conjunction with an external ke *hdbnsutil*. a. Log on to the SAP HANA server as operating syste b. Back up the new keys with the following command: `hdbnsutil -backupRootKeys <path_filena` ▪ *<dbid>* is the tenant database ID. ▪ The *<type>* option is the root key type and The value `ALL` specifies that root keys of al key types are backed up. ▪ *<path_filename>* specifies the path where extension is optional). The file is written to

2. Save the root key backup file to a secure location.

> ⚠ **Caution**
>
> Store the root key backup file in a safe location. If this file is lost, it may not be possible to recover the database.

3. Optional: To ensure that the backup file can be recovered, validate the password for the root key backup file.

> ℹ **Note**
>
> Each root key backup file created is unique. You must validate it as described here and not through comparison with other files (if multiple backups have been done).

To validate the backup file log on to the SAP HANA server as operating system user *<sid>*adm, and use the following command in the `hdbnsutil` tool:

```
hdbnsutil -validateRootKeysBackup <path_filename> [--password=<passphrase>]
```

> → **Recommendation**
>
> We recommend that you do not enter the password on the command line. You will be interactively prompted to enter it. In this way, you avoid unintentionally leaving the password in the command history and making it visible in process monitoring tools provided by the operating system.

Related Information

Encryption Configuration Control
Root Key Backup
ALTER SYSTEM SET ENCRYPTION ROOT KEYS BACKUP PASSWORD Statement (System Management)
BACKUP ENCRYPTION ROOT KEYS Statement (Backup and Recovery)
ENCRYPTION_ROOT_KEYS_EXTRACT_KEYS Function (Security)
Back Up Root Keys (SAP HANA Cockpit)

Activate Root Keys

Activate new encryption root keys so that they can be used to encrypt new data.

Prerequisites

- You are connected to the tenant database requiring the root key change.
- You have the system privilege `ENCRYPTION ROOT KEY ADMIN`.
- You have backed up the new encryption root keys. You can verify whether or not root keys are backed up by querying the system view `ENCRYPTION_ROOT_KEYS`.

Procedure

Activate the new root keys by executing the following SQL statements:

Encryption Service	Statement
Data volume encryption	`ALTER SYSTEM PERSISTENCE ENCRYPTION ACTIVATE NEW ROOT KEY`

Encryption Service	Statement
Redo log encryption	`ALTER SYSTEM LOG ENCRYPTION ACTIVATE NEW ROOT KEY`
Data and log backup encryption	`ALTER SYSTEM BACKUP ENCRYPTION ACTIVATE NEW ROOT KEY`
Internal application encryption	`ALTER SYSTEM APPLICATION ENCRYPTION ACTIVATE NEW ROOT KEY`

If encryption is enabled, new data is encrypted with the new root keys.

i Note

It is not necessary to enable the internal application encryption service explicitly. It is available automatically to requesting applications.

Related Information

Generate New Root Keys
Back Up Root Keys
ENCRYPTION_ROOT_KEYS System View
ALTER SYSTEM PERSISTENCE ENCRYPTION Statement (System Management)
ALTER SYSTEM LOG ENCRYPTION Statement (System Management)
ALTER SYSTEM BACKUP ENCRYPTION Statement (System Management)
ALTER SYSTEM APPLICATION ENCRYPTION Statement (System Management)

Enable Encryption

You can enable data volume encryption, redo log encryption, and encryption of data and log backups in a new SAP HANA database or in an existing operational database.

Prerequisites

- To enable encryption for a tenant database, you know whether encryption configuration is controlled by the tenant database or the system database:
    - If the tenant database controls encryption configuration, encryption can only be enabled or disabled directly in the tenant database and not from the system database.
    - If the system database controls encryption configuration, encryption can only be enabled or disabled using SQL from the system database, with a user that has the system privilege `DATABASE ADMIN`.

  See *Encryption Configuration Control*.
- You have the system privilege `ENCRYPTION ROOT KEY ADMIN`.
- If necessary, you have changed and backed up the encryption root keys. See *Changing Encryption Root Keys*.

Context

It is recommended that you enable encryption in the system database and the tenant databases when they are created. In this way, you ensure that all the pages are encrypted. If you received SAP HANA from a hardware or hosting partner, you should enable

encryption after handover and before importing your sensitive data.

If you enable encryption in an operational database, only the pages in use in the data volumes are encrypted. Pages in data volumes that are not in use may still contain old content, and are only overwritten and encrypted over time. This means that your data in data volumes will only be fully encrypted after some delay. In addition, only redo log entries that are created after encryption is enabled are encrypted. Redo log files that were created before encryption was enabled are not encrypted. Although encryption can be switched on at any point in time, unencrypted data can remain on disk. If this is not wanted, you will need to install a new database on a fresh hard drive, activate encryption, import a backup, and low-level erase the old disks.

You can enable encryption of full data backups, delta data backups, and log backups in the database at any time.

→ Recommendation

Although SAP HANA provides you with the flexibility to encrypt data volumes, redo logs, and backups independently of each other, if you require full protection in the persistence layer, we recommend that you enable all services.

Procedure

ⓘ Note

In a system-replication configuration, enable (or disable) encryption in the primary system only. The setting will be propagated to all secondary systems. The secondary systems must be running and replicating.

Enable the required encryption service using the SAP HANA cockpit (tenant database control only) or the following SQL statements:

If the tenant database has control	• Data volume encryption
	`ALTER SYSTEM PERSISTENCE ENCRYPTION ON`
	• Redo log encryption
	`ALTER SYSTEM LOG ENCRYPTION ON`
	• Backup encryption
	`ALTER SYSTEM BACKUP ENCRYPTION ON`

If the system database has control	• Data volume encryption
	`ALTER DATABASE <database_name> PERSISTENCE ENCRYPTION ON`
	• Redo log encryption
	`ALTER DATABASE <database_name> LOG ENCRYPTION ON`
	• Backup encryption
	`ALTER DATABASE <database_name> BACKUP ENCRYPTION ON`

Results

Data volume encryption and redo log encryption

All data persisted to data volumes is encrypted and all future redo log entries persisted to log volumes are encrypted.

Backup encryption

Backup encryption is enabled. Subsequent log backups, as well as full backups and delta data backups will be encrypted.

> **Note**
> If backup encryption is active, a data snapshot is **not automatically encrypted**. For more information, see *Backup Encryption*.

Related Information

Encryption Configuration
Encryption Configuration Control
Changing Encryption Root Keys
Data and Log Volume Encryption
Backup Encryption
Enable Encryption (SP HANA Cockpit)
M_ENCRYPTION_OVERVIEW System View
ALTER SYSTEM PERSISTENCE ENCRYPTION Statement (System Management)
ALTER SYSTEM LOG ENCRYPTION Statement (System Management)
ALTER DATABASE Statement (Tenant Database Management)
ALTER SYSTEM BACKUP ENCRYPTION Statement (System Management)
SAP Note 2159014

Disable Encryption

Disabling data volume encryption triggers the decryption of all encrypted data. Newly persisted data is not encrypted. Disabling redo log encryption makes sure that future redo log entries are not encrypted when they are written to disk.

Prerequisites

- To disable encryption for a tenant database, you know whether encryption configuration is controlled by the tenant database or the system database:
  - If the tenant database controls encryption configuration, encryption can only be enabled or disabled directly in the tenant database and not from the system database.
  - If the system database controls encryption configuration, encryption can only be enabled or disabled using SQL from the system database, with a user that has the system privilege `DATABASE ADMIN`.

  See *Encryption Configuration Control*.
- You have the system privilege `ENCRYPTION ROOT KEY ADMIN`.

Procedure

> **Note**

In a system-replication configuration, enable (or disable) encryption in the primary system only. The setting will be propagated to all secondary systems. The secondary systems must be running and replicating.

Disable the required encryption service using the SAP HANA cockpit (tenant database control only) or the following SQL statements:

If the tenant database has control	• Data volume encryption
	`ALTER SYSTEM PERSISTENCE ENCRYPTION OFF`
	• Redo log encryption
	`ALTER SYSTEM LOG ENCRYPTION OFF`
	• Backup encryption
	`ALTER SYSTEM BACKUP ENCRYPTION OFF`

If the system database has control	• Data volume encryption
	`ALTER DATABASE <database_name> PERSISTENCE ENCRYPTION OFF`
	• Redo log encryption
	`ALTER DATABASE <database_name> LOG ENCRYPTION OFF`
	• Backup encryption
	`ALTER DATABASE <database_name> BACKUP ENCRYPTION OFF`

Results

Data volume encryption

Data starts being decrypted in the background. Depending on the size of the SAP HANA database, this process can be very time consuming. Only after this process has completed is all your data decrypted. Newly persisted data is not encrypted.

Redo log encryption

New redo log entries are not encrypted. Existing redo log entries are not decrypted. Log entries will only be fully unencrypted when all encrypted entries have been overwritten.

Backup encryption

New data backups, delta backups, and log backups are not encrypted. On an unencrypted data volume, data snapshots are also unencrypted.

Related Information

Encryption Configuration
Encryption Configuration Control
Disable Encryption (SP HANA Cockpit)

ALTER SYSTEM LOG ENCRYPTION Statement (System Management)
ALTER SYSTEM BACKUP ENCRYPTION Statement (System Management)
ALTER DATABASE Statement (Tenant Database Management)

Import Backed-Up Root Keys Before Database Recovery

Before performing a recovery from encrypted data and log backups, you must import the backed-up root keys.

Prerequisites

- You have the credentials of the operating system user (`<sid>adm`).
- You can log on to the system database and have the system privilege `DATABASE STOP`.
- The location of the root key backup file is accessible.
- If using SQL to import keys: You have the system privilege `DATABASE RECOVERY OPERATOR` or `DATABASE ADMIN` and `ENCRYPTION ROOT KEY ADMIN` in the system database.
- If using *hdbnsutil* to import keys: You know the ID of the database whose root keys you want to import. You can determine the IDs of all tenant databases by executing the following SQL command in the system database:

```
SELECT DATABASE_NAME,
        CASE WHEN (DBID = '' AND
        DATABASE_NAME = 'SYSTEMDB')
        THEN 1
        WHEN (DBID = '' AND
        DATABASE_NAME <> 'SYSTEMDB')
        THEN 3
        ELSE TO_INT(DBID)
        END DATABASE_ID
FROM (SELECT DISTINCT DATABASE_NAME, SUBSTR_AFTER (SUBPATH,'.') AS DBID FROM SYS_DATABASES.M_
```

Context

Before you recover an encrypted database, you must first import backed-up root keys to initialize the secure store. When the instance SSFS (secure store in the file system) is used as the secure store, the set of keys from the root key backup file replaces the set of keys in the instance SSFS.

If the LSS is used with an external KMS (key management system), backed-up root keys can also be recovered using a second access key pair in an emergency situation. This is only necessary if the private key stored in the KMS has been lost, damaged, or becomes unusable for some reason.

You can import root keys in the following ways:

Option	Execution Location	Database Mode	Scope
SQL statement `RECOVER ENCRYPTION ROOT KEYS`	System database	The tenant database must be offline.	Encryption root keys can only be recovered for tenant databases.

Option	Execution Location	Database Mode	Scope
hdbnsutil* tool	SAP HANA server	The tenant database must be offline.	Encryption root keys can be recovered for tenant databases and system databases.

ℹ Note

*You cannot recover root keys using *hdbnsutil* if LSS is used in conjunction with an external KMS.

Procedure

1. Log on to the SAP HANA server as operating system user *<sid>*adm.
2. Validate that you have the password for the root key backup file.

   To validate the SSFS format root key backup file format, execute:

   ```
   cd /usr/sap/<sid>/<HDBinstance_no>/exe
   ./hdbnsutil -validateRootKeysBackup <filename> [--password=<password>]
   ```

3. In the system database, stop the tenant database to be recovered.

   You can do this in the SAP HANA cockpit or by executing the statement `ALTER SYSTEM STOP DATABASE <database_name>`.

4. Import backed-up root keys using one of the following methods:
   - Using the SQL statement `RECOVER ENCRYPTION ROOT KEYS`. The input backup file format can be SSFS format root keys backup.
     - To recover keys from the SSFS (not connected to an external KMS), execute:

       ```
       RECOVER ENCRYPTION ROOT KEYS (<root_keytype_list>) FOR <database_name> USING <root
       ```

       - *<database_name>* is the name of the tenant database.
       - *<root_keytype_list>* (optional) specifies the root key types and accepts the values `PERSISTENCE`, `LOG`, `BACKUP`, and `APPLICATION`. If you do not specify any value for *<root_keytype_list>*, all root key types are imported.
       - *<root_key_backup_definition_file>* specifies the SSFS format root key backup in the file system.

         ### ℹ Note

         If an LSS format backup file is provided the statement will throw an exception.

       - *<password>* is the root key backup password.
   - Using the `hdbnsutil` program

     ### ℹ Note

     You cannot recover a root key backup using *hdbnsutil* if the LSS is used in conjunction with an external KMS.

     Execute the following command to recover the SSFS format root key backup:

     ```
     cd /usr/sap/<sid>/<HDBinstance_no>/exe
     ./hdbnsutil -recoverRootKeys <filename> --dbid=<dbid> --password=<password> --type=ALL
     ```

     - *<dbid>* is the tenant database ID.

- *<type>* import the backed-up root keys using either the option is the root key type and also accepts the values `PERSISTENCE`, `LOG`, `BACKUP`, and `APPLICATION`. The value `ALL` specifies that root keys of all types are imported. If you do not specify any value for *<type>*, all key types are imported.
- *<password>* is the root key backup password.

→ Recommendation

We recommend that you do not enter the password on the command line (`--password`). By omitting this parameter, you will be interactively prompted to enter it.

ℹ Note

If you have backed-up root keys to different files, for example according to root key type, you need to execute the command several times.

Related Information

Stop a Tenant Database
Prerequisites: Recovering an Encrypted SAP HANA Database
Using the LSS with an External Key Management System
Recovering an SAP HANA Database
RECOVER ENCRYPTION ROOT KEYS Statement (Backup and Recovery)