PUBLIC

SAP HANA Platform 2.0 SPS 00
Document Version: 1.0 – 2016-11-30

SAP HANA Multitenant Database Containers

Operations Guide

Content

1 Getting Started. 5

2 Overview of SAP HANA Multitenant Database Containers. 6

2.1 Server Architecture of Multiple-Container Systems. 7

2.2 Scale-Out Architecture of Multiple-Container Systems. 8

2.3 The System Database. 10

2.4 Cross-Database Access. 11

2.5 Database Isolation. 12

2.6 Administration of Multitenant Database Containers. 14

2.7 Configuration Parameters in Multiple-Container Systems. 15

2.8 System and Statistics Views in Multiple-Container Systems. 19

2.9 Connections for Multitenant Database Containers. 20

3 Security and Multitenant Database Containers. 25

3.1 SYSTEM User in Multitenant Database Containers. 27

3.2 Cross-Database Authorization in Multitenant Database Containers. 28

3.3 Data and Log Volume Encryption in Multitenant Database Containers. 30

3.4 Restricted Features in Multitenant Database Containers. 31

3.5 Default Blacklisted System Properties in Multitenant Database Containers. 34

4 Creating and Configuring Tenant Databases. 36

4.1 Converting an SAP HANA System to Support Multitenant Database Containers. 37

Convert to Multitenant Database Containers Using the Graphical User Interface. 37

Convert to Multitenant Database Containers Using the Command-Line Interface. 39

Convert to Multitenant Database Containers Using the Web User Interface. 41

Parameter Reference: Converting an SAP HANA System to Support Multitenant Database Containers. 45

4.2 Increase the System Isolation Level. 46

File and Directory Permissions with High Isolation. 48

4.3 Decrease the System Isolation Level. 49

4.4 Create a Tenant Database. 50

4.5 Disable Features on a Tenant Database. 53

4.6 Enable and Configure Cross-Database Access. 54

Workload Management and Cross-Database Queries. 56

Troubleshooting Error Situations Related to Cross-Database Access. 57

4.7 Prevent Changes to System Properties in Tenant Databases. 60

4.8 Configure HTTP(S) Access to Multitenant Database Containers via SAP HANA XS Classic. 62

5 Monitoring and Managing Tenant Databases. 67
5.1 Monitoring Multitenant Resources in SAP HANA Cockpit. 68
Monitor Status and Resource Usage of a Multitenant Resource. 68
Monitor Alerts for a Multitenant Resource. 75
Stop and Start a Tenant Database. 79
Delete a Tenant Database. 80
5.2 Add a Service to a Tenant Database. 82
5.3 Remove a Service from a Tenant Database. 84
5.4 View Diagnosis Files of an Unavailable Tenant Database. 85
5.5 Resetting the SYSTEM User Password in Multitenant Database Containers. 87
Reset the SYSTEM User Password of the System Database. 87
Reset the SYSTEM User Password of a Tenant Database. 89

6 Configuring Memory and CPU Usage in Multiple-Container Systems. 91
6.1 Parameter Reference: Memory Consumption. 92
6.2 Controlling Parallel Execution of SQL Statements 93
6.3 Controlling CPU Consumption. 95

7 Copying and Moving Tenant Databases Between Systems. 99
7.1 Copy and Move Process. 103
7.2 Security of the Copy and Move Process. 107
Disable Secure Network Communication. 109
7.3 Preparing to Copy or Move a Tenant Database. 109
Verify TLS/SSL Configuration of Internal Communication Channels. 110
Open Communication From Target to Source System. 111
Set Up Trust Relationship Between Target and Source Systems. 112
Create Credential for Authenticated Access to Source System. 113
Back Up Tenant Database. 114
7.4 Copy a Tenant Database to Another System. 115
7.5 Move a Tenant Database to Another System. 117
7.6 Perform Manual Post-Copy/Move Tasks. 119
7.7 Tutorial: Moving a Tenant Database. 120
Verify TLS/SSL Configuration of Internal Communication Channels in M01 and M02. 122
Open Communication From M02 to M01. 123
Configure TLS/SSL for External Communication in M01 and M02. 124
Set Up Trust Relationship Between M02 and M01 127
Create Credential for Authenticated Access from M02 to M01. 130
Back Up Tenant Database NEW_APP_TEST. 131
Create OS Group and OS for New Tenant Database in M02. 132
Move Tenant Database NEW_APP_TEST. 132

8 Using SAP Web Dispatcher for Load Balancing with Tenant Databases. 134

8.1 Virtual-Host-Based Routing. 135
8.2 Configuring an External SAP Web Dispatcher for Tenant Databases. 137
Option 1: Configuring Access to Multiple (or All) Tenant Databases Through External Web Dispatcher Only. 139
Option 2: Configuring Access to Multiple (or All) Tenant Databases Through External Web Dispatcher and Directly. 142
Additional Configuration Required in SPS 09 and SPS 10. 145

9 SAP HANA System Replication with Multitenant Database Containers. 149

10 Backing Up and Recovering Multitenant Database Containers. 151

11 Important Disclaimer for Features in SAP HANA Platform, Options and Capabilities. 153

1 Getting Started

SAP HANA multitenant database containers represent the basis for multitenancy in SAP HANA. Being able to run and manage multiple tenant databases in one SAP HANA system helps you to lower capital expenditure, simplify database management and build multi-tenant cloud applications.

This guide covers configuration, administrative, monitoring, and other operating tasks typically performed by the administrator of an SAP HANA system that supports multitenant database containers. For more information about how to install a multiple-container system, see the *SAP HANA Server Installation and Update Guide*.

> **Note**
>
> If you have SAP HANA options installed, review the section about multitenant database containers in the administration guide of the corresponding option for additional information before proceeding. Be aware that you need additional licenses for SAP HANA options and capabilities. For more information, see *Important Disclaimer for Features in SAP HANA Platform, Options and Capabilities*.

Important SAP Notes

Table 1:

SAP Note Number	Title
2096000	SAP HANA Multitenant Database Containers – Additional Information
2101244	FAQ: SAP HANA Multitenant Database Containers

Related Information

SAP HANA Server Installation and Update Guide (PDF)
SAP HANA Server Installation and Update Guide (HTML)
Important Disclaimer for Features in SAP HANA Platform, Options and Capabilities [page 153]

8.13

2 Overview of SAP HANA Multitenant Database Containers

SAP HANA supports multiple isolated databases in a single SAP HANA system. These are referred to as multitenant database containers.

An SAP HANA system installed in multiple-container mode is capable of containing more than one multitenant database containers. Otherwise, it is a single-container system. Single-container systems can be converted to multiple-container systems.

A multiple-container system always has exactly one system database, used for central system administration, and any number of multitenant database containers (including zero), also called tenant databases. An SAP HANA system installed in multiple-container mode is identified by a single system ID (SID). Databases are identified by a SID and a database name. From the administration perspective, there is a distinction between tasks performed at system level and those performed at database level. Database clients, such as the SAP HANA studio, connect to specific databases.

All the databases in a multiple-container system share the same installation of database system software, the same computing resources, and the same system administration. However, each database is self-contained and fully isolated with its own:

- Set of database users
- Database catalog
- Repository
- Persistence
- Backups
- Traces and logs

Although database objects such as schemas, tables, views, procedures, and so on are local to the database, cross-database SELECT queries are possible. This supports cross-application reporting, for example.

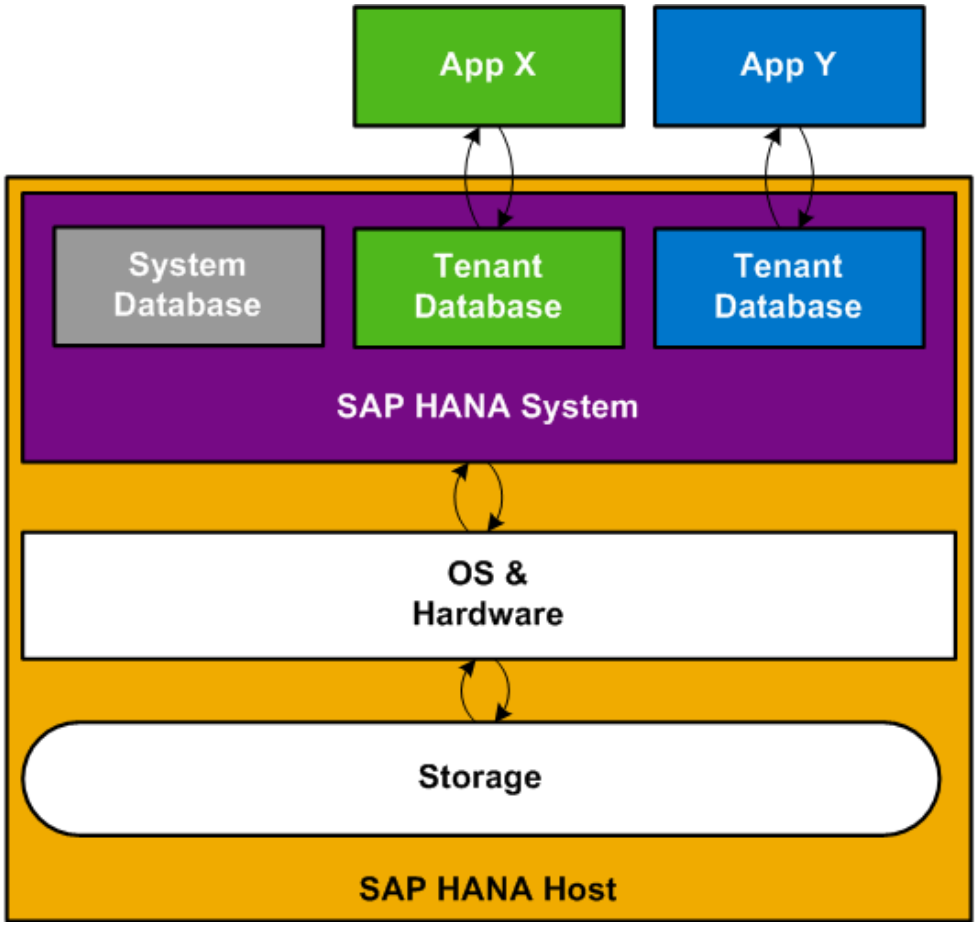


Figure 1: On-premise Deployment Using SAP HANA Multitenant Database Containers

2.1 Server Architecture of Multiple-Container Systems

An SAP HANA system consists of multiple servers: name server, index server, preprocessor server, XS server and so on. The databases in a multiple-container system run different combinations of these servers.

Only the **system database** runs the name server. The name server contains landscape information about the system as a whole, including which tenant databases exist. It also provides indexserver functionality for the system database. Unlike the name server in a single-container system, the name server of the system database in a multiple-container system does not own topology information, that is, information about the location of tables and table partitions in databases. Database-related topology information is stored in the relevant tenant database catalog.

8.13 **Tenant databases** require only an own index server. Servers that do not persist data, such as the compile server and the preprocessor server, run on the system database and serve all databases.

The following figure shows a sample multiple-container system with three databases (system database and two tenant databases) on a single host.

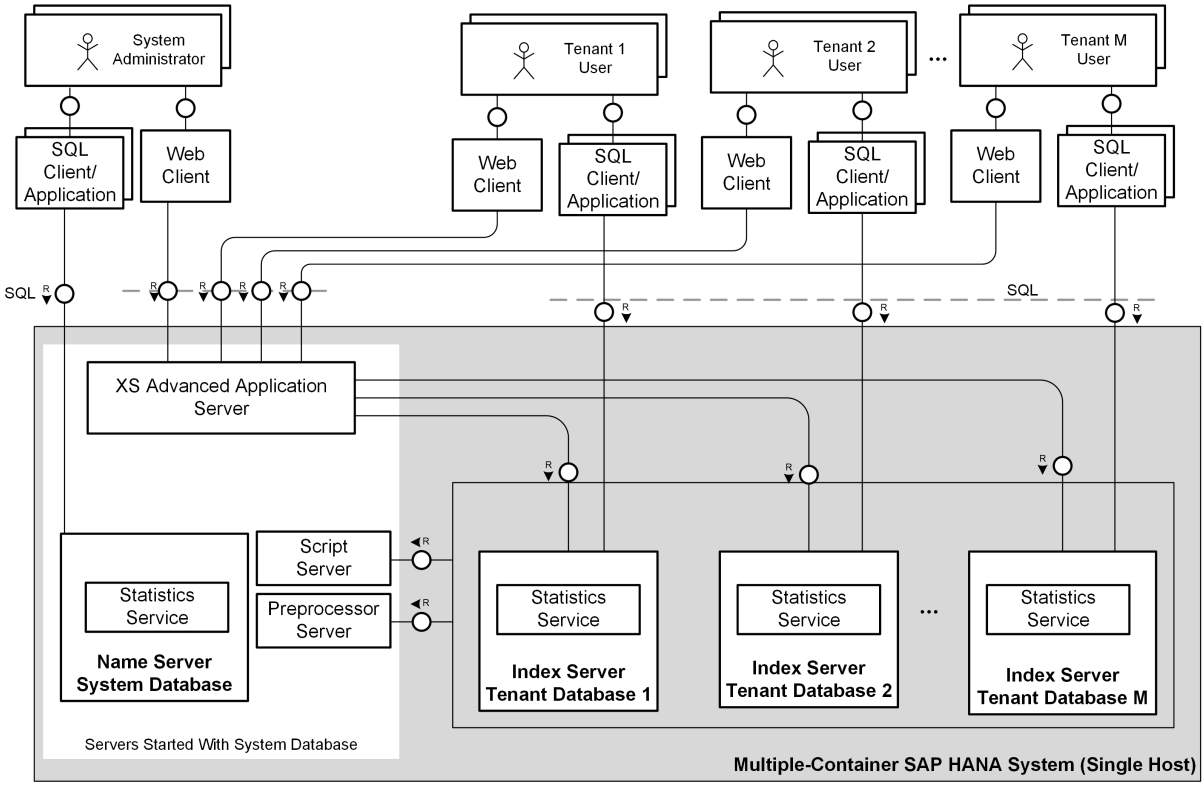


Figure 2: Single-Host SAP HANA System with Multitenant Database Containers

> **Note**
>
> If the SAP HANA XS classic server is available, it runs embedded in the (master) index server of the tenant database by default, although it can be added as a separate service if necessary. The **SAP Web Dispatcher**, which runs as a separate database service on the system database, is used to route incoming HTTP requests from clients to the correct XS classic server based on virtual host names. This is part of network configuration. In addition to the system-internal Web Dispatcher, you can implement an external Web Dispatcher for load distribution. See *Using SAP Web Dispatcher for Load Balancing with Tenant Databases*.

Related Information

Using SAP Web Dispatcher for Load Balancing with Tenant Databases [page 134]
Scale-Out Architecture of Multiple-Container Systems [page 8]

2.2 Scale-Out Architecture of Multiple-Container Systems

It is possible to distribute tenant databases across several hosts (scale-out system).

To ensure system availability, an instance of the system database runs on all hosts (worker and standby) in a single master and multiple workers configuration. Tenant databases can be created on worker hosts and

existing databases can be scaled out through the addition of services. If a host fails, the standby instance will fail over all active databases and their services.

The following figure shows a multiple-container system with three tenant databases distributed across three hosts.

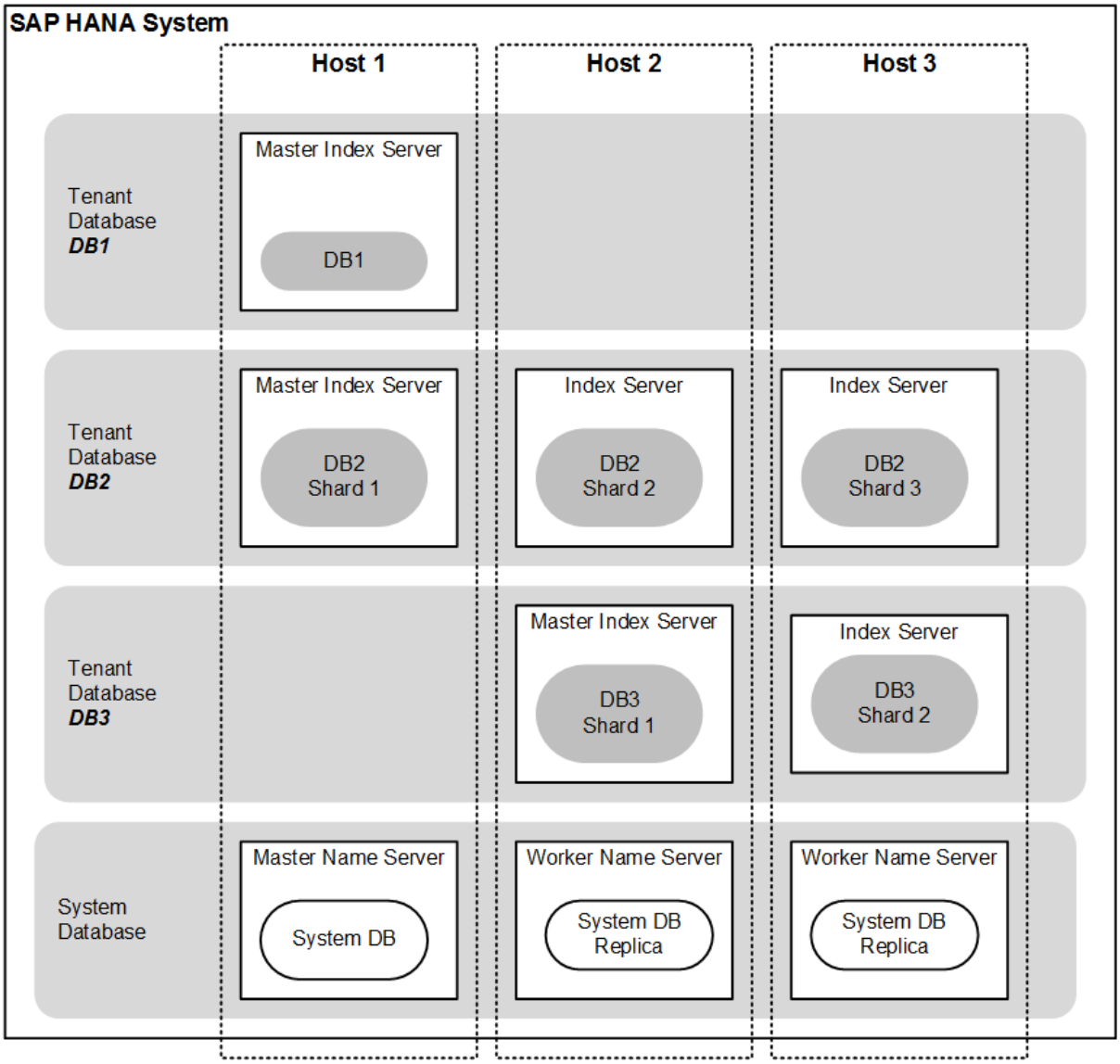


Figure 3: Multiple-Host System with Multitenant Database Containers

Scale-Out Recommendations

When planning an SAP HANA MDC deployment, various options exist with regard to scale-up versus scale-out.

In general, scaling up offers some performance advantages over scaling out, as memory access is local and minor overhead associated with inter-node network communication is avoided.

Note the following with regard to scale-out:

- It is possible to distribute tenant databases across several hosts in a scale-out system.

- The primary reason to distribute tenant databases generally is when their size is larger than the capacity of a single host. However, other reasons for distributing tenant database may exist, for example, a large SAP Business Warehouse (BW) system requires a scale-out configuration in accordance with its sizing rules.
- If tenant databases are distributed in a scale-out configuration due to sizing requirements, caution is advised when deploying additional tenant databases on the same host as a distributed tenant database shard. The rationale is this: Workload in distributed scenarios can be somewhat volatile and less predictable. Therefore in many cases, it can be advantageous to dedicate maximum resources of the host to the distributed tenant database shard in order to maintain expected performance.
- In certain cases, more than one distributed tenant database shard may share the same host. In these cases, in order to dedicate maximum resources for a master node (for performance reasons), it is advisable to avoid deploying other tenant databases on the master node. For example, the following deployment should offer performance advantages:
  - Host 1: Master for tenant database 1
  - Host 2: Worker for tenant database 1 and worker for tenant database 2
  - Host 3: Master for tenant database 2
  - Host 4: Standby host for failover

2.3 The System Database

The system database is created during either installation of a multiple-container system or conversion from a single-container system to a multiple-container system. The system database contains information about the system as a whole, as well as all its tenant databases. It is used for central system administration.

A multiple-container system has exactly one system database. It is created during system installation or migration from a single-container system. It contains the data and users for system administration. System administration tools, such as the SAP HANA cockpit or the SAP HANA studio, can connect to this database. The system database stores overall system landscape information, including knowledge of the tenant databases that exist in the system. However, it doesn't own database-related topology information, that is, information about the location of tables and table partitions in databases. Database-related topology information is stored in the relevant tenant database catalog.

Administration tasks performed in the system database apply to the system as a whole and all of its databases (for example, system-level configuration settings), or can target specific tenant databases (for example, backup of a tenant database). For more information, see *Administration of Multitenant Database Containers*.

Things to Remember About the System Database

- The system database is not a database with full SQL support.
- The system database cannot be distributed across multiple hosts, in other words, scale-out is not possible.
- If you need a full-featured SAP HANA database in a multiple-container system, you always have to create at least one tenant database.
- The system database can show monitoring data from tenant databases (views in the schema SYS_DATABASES) but can never show actual content from tenant databases.

Related Information

Administration of Multitenant Database Containers [page 14]
Configuring Memory and CPU Usage in Multiple-Container Systems [page 91]

2.4 Cross-Database Access

Read-only queries between tenant databases in the same SAP HANA system are possible. This supports cross-application reporting. Cross-database access must be explicitly enabled.

Every tenant database in a multiple-container system is self-contained with its own isolated set of database users and isolated database catalog. However, to support for example cross-application reporting, cross-database SELECT queries are possible. This means that database objects such as tables and views can be local to one database but be read by users from other databases in the same system.

The following object types on remote databases can be accessed using cross-database access:

- Schemas
- Rowstore and columnstore tables (not including virtual tables)
- SQL views (not including monitoring views)
- Graphical calculation views
  - If they only use supported object types as data sources
  - If they don't use procedure-based analytic privileges
- Synonyms

The following object types on the local tenant database can access database objects on the remote tenant database:

- SQL views
- Scripted and graphical calculation views
- Procedures
- Synonyms

The SAP HANA modeler supports modeling of graphical calculation views using tables and other graphical calculation views as data sources from different tenant databases. For more information, see *Multitenant Database Containers Support for Modeling Graphical Calculation Views* in the *SAP HANA Modeling Guide (For SAP HANA Studio)*.

For more information about how to enable and configure cross-database access, see *Enable and Configure Cross-Database Access*.

Related Information

Enable and Configure Cross-Database Access [page 54]
Cross-Database Authorization in Multitenant Database Containers [page 28]
Troubleshooting Error Situations Related to Cross-Database Access [page 57]

Workload Management and Cross-Database Queries [page 56]
SAP HANA Modeling Guide (For SAP HANA Studio)

2.5 Database Isolation

Every tenant database in a multiple-container system is self-contained and isolated in terms of users, database catalog, repository, logs, and so on. However, to protect against unauthorized access at the operating system (OS) level, it's possible to increase isolation further through OS user separation and authenticated communication within databases.

OS User Separation

By default, all database processes in a multiple-container system run under the default OS user `<sid>`adm. If it's important to mitigate against cross-database attacks through OS mechanisms, you can configure the system for high isolation. In this way, the processes of individual tenant databases must run under dedicated OS users belonging to dedicated OS groups, instead of all database processes running under `<sid>`adm. Database-specific data on the file system is subsequently protected using standard OS file and directory permissions.

> **Note**
>
> `<sid>`adm is the OS user for the system database.

Authenticated Communication

In addition, once high isolation has been configured, internal database communication is secured using the Transport Layer Security (TLS)/Secure Sockets Layer (SSL) protocol. Certificate-based authentication is used to ensure that only the processes belonging to the same database can communicate with each other. It is also possible to configure internal communication so that all data communication within databases is encrypted.

> **Note**
>
> If cross-database access is enabled, communication between configured tenant databases is allowed.

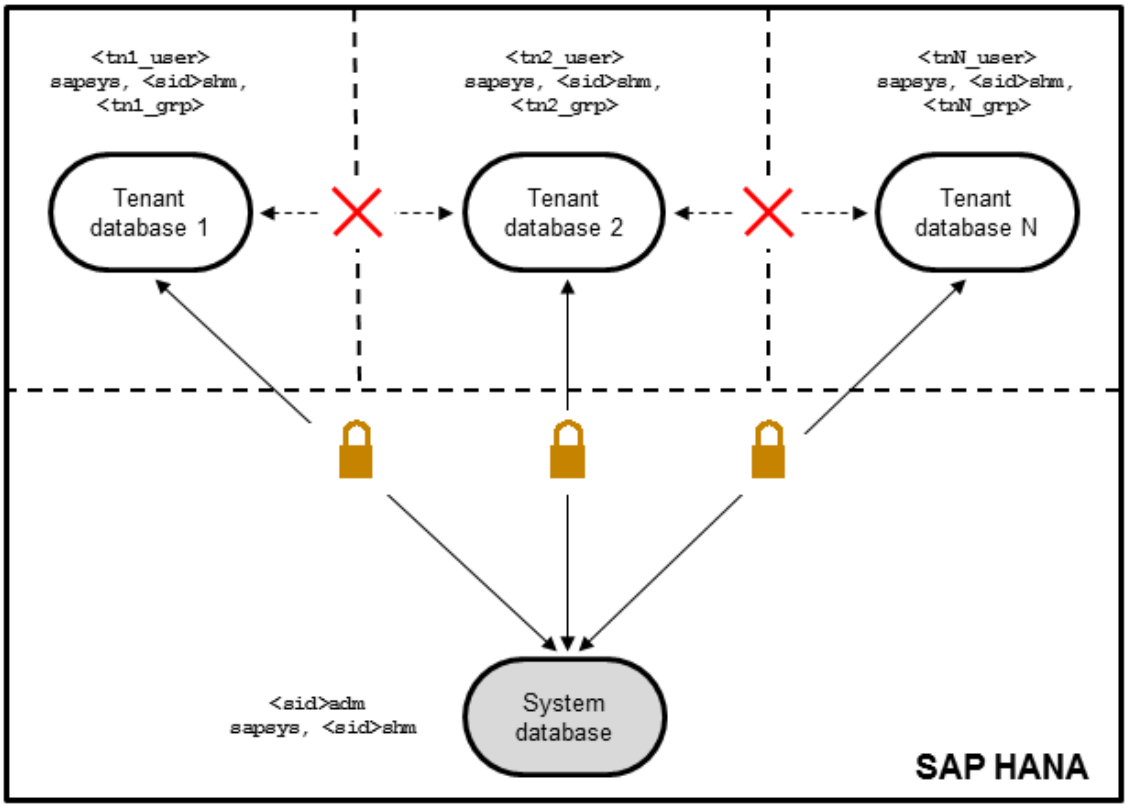


Figure 4: High Database Isolation

Configuration

You can specify the isolation level of the system during installation. The default isolation level is low. It is also possible to change the isolation level of an existing system (from low to high or from high to low) at any time. Once high isolation has been configured, a dedicated OS user and group must exist for every tenant database. Otherwise, it's not possible to create or start a tenant database.

Internal database communication is secured with the same mechanism used for securing other internal SAP HANA communication channels. Once high isolation has been configured, authenticated communication within databases is enabled without any change required to the default TLS/SSL configuration for internal communication. However, encryption of data communication may need to be configured explicitly.

For more information, see:

- *Installing a Multitenant Database Container SAP HANA System* in the *SAP HANA Server Installation and Update Guide*
- *Increase the System Isolation Level* in the *SAP HANA Administration Guide*
- *Secure Internal Communication* in the *SAP HANA Security Guide*

Related Information

Increase the System Isolation Level [page 46]
File and Directory Permissions with High Isolation [page 48]
SAP HANA Server Installation and Update Guide
SAP HANA Security Guide
SAP HANA Administration Guide

2.6 Administration of Multitenant Database Containers

In SAP HANA systems that support multitenant database containers, there is a distinction between administration tasks performed at system level and those performed at database level.

System Versus Database Administration

Unlike a single-container system in which system and database are perceived as a single unit and are therefore administered as one, multiple-container systems have two levels of administration.

Some administration tasks are performed in the system database and apply globally to the system and all its databases. They include for example:

- Starting and stopping the whole system
- Monitoring the system
- Configuring parameters in configuration (*ini) files at system level
- Setting up and configuring tenant databases, for example:
  - Creating and dropping tenant databases
  - Disabling features on tenant databases
  - Configuring system- and database-specific parameters in configuration (*ini) files
  - Scaling out tenant databases by adding services
- Backing up tenant databases
- Recovering tenant databases

Some administration tasks are performed in the tenant database and apply only to that database. They include for example:

- Monitoring the database
- Provisioning database users
- Creating and deleting schemas, tables, and indexes in the database
- Backing up the database
- Configuring database-specific parameters in configuration (*ini) files

Administration Tools

Several tools are available for the administration of SAP HANA. While all tools support database-level administration (which is comparable to the administration of a single-container system), system-level administration of tenant databases requires the SAP HANA cockpit (for example, monitoring availability of tenant databases, creating and deleting tenant databases). The SAP HANA studio is required for system configuration tasks.

For more information about the SAP HANA cockpit and other administration tools, see *SAP HANA Administration Tools* in the *SAP HANA Administration Guide*.

Related Information

The System Database [page 10]
Creating and Configuring Tenant Databases [page 36]
Monitoring and Managing Tenant Databases [page 67]

2.7 Configuration Parameters in Multiple-Container Systems

In addition to the layers "default", "system", and "host", system configuration files in multiple-container systems have a "database" layer to facilitate the configuration of properties for individual databases.

Database-Specific Configuration

In general, you can configure database-specific properties both in the system database and in tenant databases themselves. Properties configured in the system database can be applied to all databases (if configured in the system layer) or to specific databases (if configured in database layer).

Properties configured in a tenant database apply to that tenant database only. Only properties in the following files can be configured in tenant databases:

- `attributes.ini`
- `dpserver.ini`
- `esserver.ini`
- `executor.ini`
- `extensions.ini`
- `global.ini`
- `indexserver.ini`
- `multidb.ini`

- `scriptserver.ini`
- `xsengine.ini`

File Location

If properties are configured in the database layer, a database-specific configuration file is stored at the following location on the server: `/hana/shared/$SID/global/hdb/custom/config/DB_<dbname>`

Example

The properties in the `nameserver.ini` file are not database specific. They can only be configured at system level. The `nameserver.ini` file is therefore stored at `/hana/shared/$SID/global/hdb/custom/config`.

However, the properties in the `indexserver.ini` can be database specific. Properties that are configured in the system layer and apply to all databases are stored in the `indexserver.ini` at `/hana/shared/$SID/global/hdb/custom/config`. Properties configured for an individual database override the system-layer value and are stored in the `indexserver.ini` at `/hana/shared/$SID/global/hdb/custom/config/DB_<dbname>`.

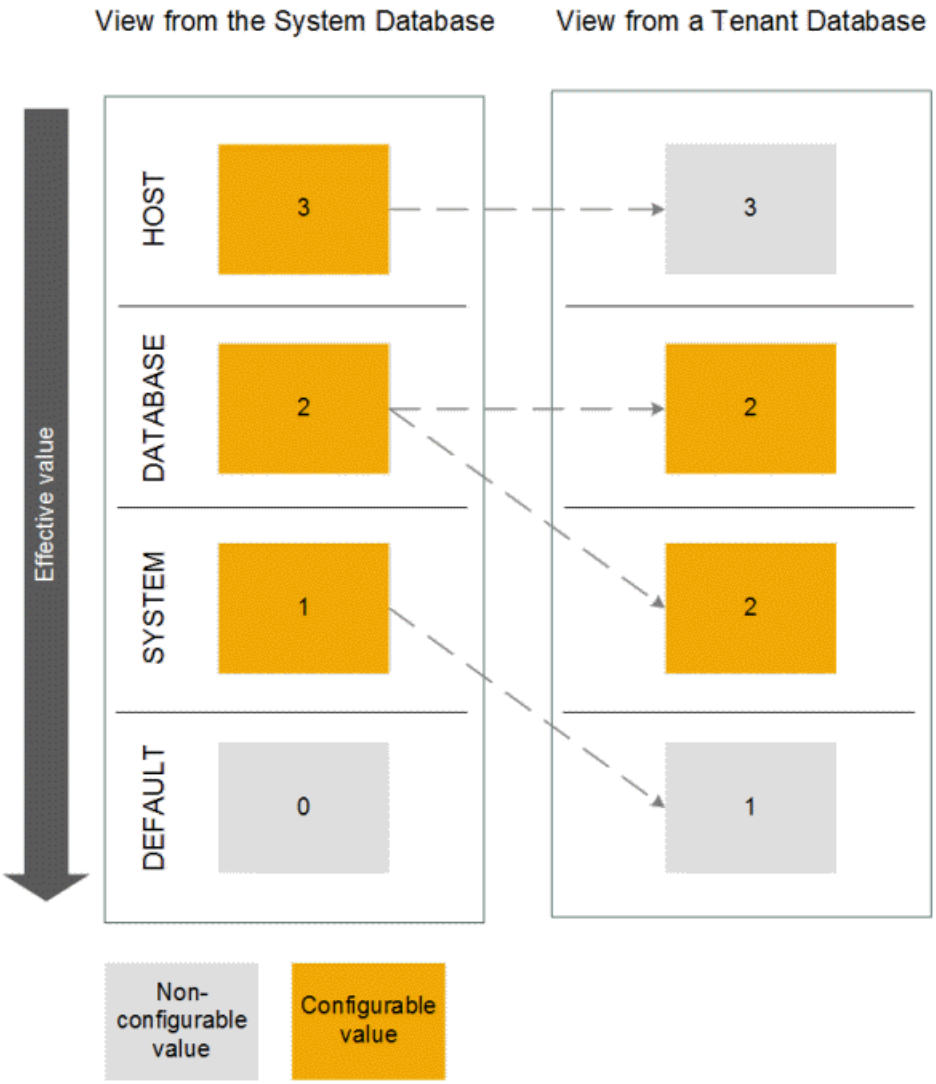
Layered Configuration

Many properties can be configured in the system, host, and database layer. Values configured in the database layer take precedence over system-layer values.

However, when you are connected to a tenant database, you will see the database-layer value of a property is also displayed as the system-layer value. This is because from the perspective of the tenant database, the database and the system are effectively the same. In addition, it allows client applications designed for single-container systems (where the system and the database **are** the same thing) to query the system-layer value and still retrieve the correct value for the tenant database. The true system-layer value (that is, the value configured for all databases in the system database) is displayed in the tenant database as the default-layer value.

Values configured in the host layer take precedence over database-layer values. Host values can only be configured in the system database.

The following figure illustrates how layered configuration work. See also the example below.

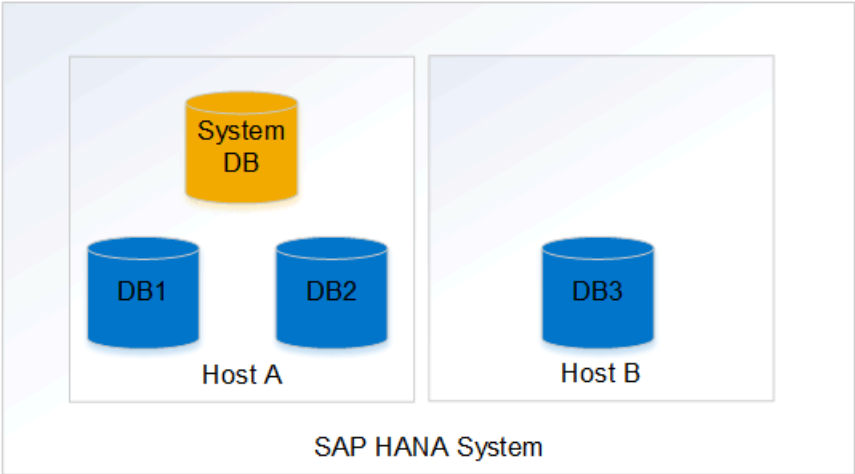


You can view actual configuration values in the Administration editor of the SAP HANA studio (*Configuration* tab) or by querying the following system views:

- M_INIFILE_CONTENTS (SYS_DATABASES)
  This view can be accessed only from the system database. It contains the values configured for all properties on system, host, and database layer for **all** active databases.
- M_INIFILE_CONTENTS (SYS)
  This view is available in every database and contains the values that apply to the database in question. Values that were configured in the system layer in the system database are identified as default-layer values. Values that were configured in the database layer in the tenant database are identified as system- and database-layer values. Values configured at the host layer are shown only for hosts on which the database is running.

Example

A multiple-container system has 3 tenant databases DB1, DB2, and DB3, distributed across 2 hosts Host A and Host B:



The default value of the property `[execution] max_concurrency` in the `global.ini` file is `0`. The system administrator changes the default configuration of this property in the `indexserver.ini file` as follows:

First, the system administrator creates a new system-layer value (**`10`**) in `indexserver.ini`. Since the system-layer value applies to all tenant databases and cannot be changed by a tenant database user, users on all tenant databases initially see the value `10` as the default configuration:

Executed in DB1, DB2, and DB3:

```
SELECT * FROM SYS.M_INIFILE_CONTENTS WHERE KEY='max_concurrency'

| FILE_NAME        | LAYER_NAME   | TENANT_NAME   | HOST       | SECTION    | KEY             | VALUE |
|------------------|--------------|---------------|------------|------------|-----------------|-------|
| global.ini       | DEFAULT      |               |            | execution  | max_concurrency | 0     |
| indexserver.ini  | DEFAULT      |               |            | execution  | max_concurrency | 10 ←┼ Effective value
```

Next, the system administrator sets a new value (**`20`**) for DB1, while leaving the configuration for DB2 and DB3 unchanged.

Executed in DB1:

```
SELECT * FROM SYS.M_INIFILE_CONTENTS WHERE KEY='max_concurrency'

| FILE_NAME        | LAYER_NAME   | TENANT_NAME   | HOST       | SECTION    | KEY             | VALUE |
|------------------|--------------|---------------|------------|------------|-----------------|-------|
| global.ini       | DEFAULT      |               |            | execution  | max_concurrency | 0     |
| indexserver.ini  | DEFAULT      |               |            | execution  | max_concurrency | 10    |
| indexserver.ini  | SYSTEM       |               |            | execution  | max_concurrency | 20 ←┐  |
| indexserver.ini  | DATABASE     |               |            | execution  | max_concurrency | 20 ←┼ Effective value
```

Executed in DB2 and DB3:

```
SELECT * FROM SYS.M_INIFILE_CONTENTS WHERE KEY='max_concurrency'

| FILE_NAME        | LAYER_NAME   | TENANT_NAME   | HOST       | SECTION    | KEY             | VALUE |
|------------------|--------------|---------------|------------|------------|-----------------|-------|
| global.ini       | DEFAULT      |               |            | execution  | max_concurrency | 0     |
| indexserver.ini  | DEFAULT      |               |            | execution  | max_concurrency | 10 ←┼ Effective value
```

> **Note**
>
> In DB1, the database-layer value is duplicated to the system layer because from the perspective of the tenant database, the database and the system are effectively the same.

Finally, the system administrator sets a new value (`15`) for host A. Since host values take precedence over database values, this changes the effective value for DB1 and DB2.

Executed in DB1:

```
SELECT * FROM SYS.M_INIFILE_CONTENTS WHERE KEY='max_concurrency'

| FILE_NAME        | LAYER_NAME   | TENANT_NAME   | HOST       | SECTION    | KEY              | VALUE |
|------------------|--------------|---------------|------------|------------|------------------|-------|
| global.ini       | DEFAULT      |               |            | execution  | max_concurrency  | 0     |
| indexserver.ini  | DEFAULT      |               |            | execution  | max_concurrency  | 10    |
| indexserver.ini  | HOST         |               | HOST A     | execution  | max_concurrency  | 15 <--- Effective value
| indexserver.ini  | SYSTEM       |               |            | execution  | max_concurrency  | 20    |
| indexserver.ini  | DATABASE     |               |            | execution  | max_concurrency  | 20    |
```

Executed in DB2:

```
SELECT * FROM SYS.M_INIFILE_CONTENTS WHERE KEY='max_concurrency'

 FILE_NAME        | LAYER_NAME   | TENANT_NAME   | HOST       | SECTION    | KEY              | VALUE |
|------------------|--------------|---------------|------------|------------|------------------|-------|
| global.ini       | DEFAULT      |               |            | execution  | max_concurrency  | 0     |
| indexserver.ini  | DEFAULT      |               |            | execution  | max_concurrency  | 10    |
| indexserver.ini  | HOST         |               | HOST A     | execution  | max_concurrency  | 15 <--- Effective value
| indexserver.ini  | SYSTEM       |               |            | execution  | max_concurrency  | 20    |
```

Executed in DB3:

```
SELECT * FROM SYS.M_INIFILE_CONTENTS WHERE KEY='max_concurrency'

| FILE_NAME        | LAYER_NAME   | TENANT_NAME   | HOST       | SECTION    | KEY              | VALUE |
|------------------|--------------|---------------|------------|------------|------------------|-------|
| global.ini       | DEFAULT      |               |            | execution  | max_concurrency  | 0     |
| indexserver.ini  | DEFAULT      |               |            | execution  | max_concurrency  | 10 <--- Effective value
```

Related Information

SAP Note 203611
SAP HANA SQL and System Views Reference

2.8 System and Statistics Views in Multiple-Container Systems

Every multitenant database container system has its own SYS and _SYS_STATISTICS schemas that contain information about that database only. For system-level monitoring, additional views are accessible in the system database: the M_DATABASES (SYS) view and the views in the SYS_DATABASES schema.

- M_DATABASES
  This view is available in the SYS schema of the system database of a multiple-container system. It provides an overview of all tenant databases in the system. Only users with the system privilege DATABASE ADMIN can see the contents of this view.

- SYS_DATABASES schema
  The views in the SYS_DATABASES schema provide aggregated information from a **subset** of the views available in the SYS and _SYS_STATISTICS schemas of all tenant databases in the system. These union views have the additional column DATABASE_NAME to make it possible to identify from which database the information is coming refers. The system views in the SYS_DATABASES schema are accessible only from the system database. To be able to view information in these views, you need the system privilege DATABASE ADMIN or CATALOG READ.

Tools such as the SAP HANA cockpit use these views to support system-level monitoring of multiple-container systems.

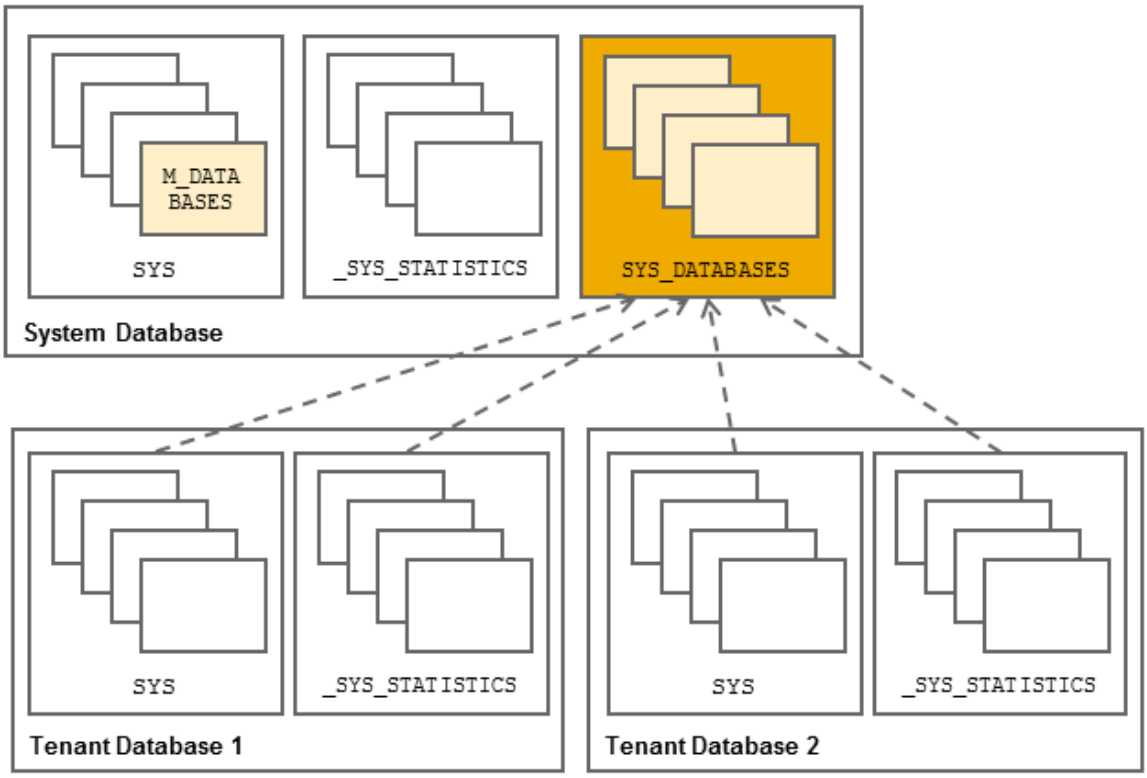


Figure 5: System and Statistics Views in Multiple-Container Systems

2.9 Connections for Multitenant Database Containers

Every tenant database in a multiple-container system has dedicated ports and connections for SQL-based client communication and internal communication.

The network communication channels used by SAP HANA can be categorized into those used for database clients connecting to SAP HANA and those used for internal communication. The *SAP HANA Administration Guide* guide provides an overview of all connection types and the standard ports used in the SAP HANA landscape. In multiple-container systems, additional ports are required to allow communication to, from and within individual tenant databases.

Port Assignment in Tenant Databases

Every tenant database in a multiple-container system has dedicated ports for SQL and internal communication. If an SAP HANA XS classic server (`xsengine` service) is running in the tenant database, there is also a dedicated port for HTTP-based client communication.

However, there are no standard port number assignments. Port numbers are assigned automatically from the available port number range according to availability at the time the database is created or a service is added. Administrators can also explicitly specify which port numbers to use when they create a tenant database or add a service.

The only exception to this is the tenant database that is automatically created when you **convert** a single-container system to a multiple-container system. This database retains the port numbers of the original single-container system: 3`<instance>`03 (internal communication), 3`<instance>`15 (SQL), and 3`<instance>`08 (HTTP via SAP HANA classic server).

The default port number range for tenant databases is 3`<instance>`40—3`<instance>`99. This means that the maximum number of tenant databases that can be created per instance is 20. However, you can increase this by reserving the port numbers of further instances. You do this by configuring the property `[multidb] reserved_instance_numbers` in the `global.ini` file. The default value of this property is 0. If you change the value to 1, the port numbers of one further instance are available (for example, 30040—30199 if the first instance is 00). If you change it to 2, the port numbers of two further instances are available (for example, 30040—30299 if the first instance is 00). And so on.

Note

The port number of the **system database** are fixed: 3`<instance>`01 (internal), 3`<instance>`13 (SQL), and 3`<instance>`14 (HTTP via XS classic server).

Let's look at some simple examples.

Example

Example 1:

You install a new SAP HANA system in multiple-container mode. Then, you create three tenant databases. Each of these tenant databases is automatically assigned port numbers for the following connection types:

- Internal communication
- SQL
- HTTP (This is the port of the XS classic server embedded in the index server.)

The first tenant database is assigned port numbers 3`<instance>`40—42, the second ports 3`<instance>`43—45, and the third 3`<instance>`46—48.

Example 2:

You install a new SAP HANA system in multiple-container mode. Then, you create a tenant database. The same port numbers as above are assigned: 3`<instance>`40 (internal communication), 3`<instance>`41 (SQL), and 3`<instance>`42 (HTTP via XS classic server). Next, you add a separate xsengine service to the first database. This service is automatically assigned the next three available port numbers: 3`<instance>`43—45. Finally, you create a second tenant database. This tenant database is automatically assigned the next three available port numbers: 3`<instance>`46—48.

Example 3:

You convert a single-container system to a multiple-container system. This results in the automatic creation of one tenant database. This tenant database has the same port numbers as the original single-container system: 3<instance>03 (internal communication), 3<instance>15 (SQL), 3<instance>08 (HTTP via XS classic server). Then, you add a second indexserver to the tenant database. It is automatically assigned port numbers 3<instance>40—42. Finally, you create a second tenant database. It is automatically assigned ports the next three available port numbers: 3<instance>43—45.

Note

All of the above examples refer to single-host systems and are based on automatic port number assignment.

You can determine the ports used by a particular tenant database by querying the M_SERVICES system view, either from the tenant database itself or from the system database.

- From the tenant database: `SELECT SERVICE_NAME, PORT, SQL_PORT, (PORT + 2) HTTP_PORT FROM SYS.M_SERVICES WHERE ((SERVICE_NAME='indexserver' and COORDINATOR_TYPE= 'MASTER') or (SERVICE_NAME='xsengine'))`
- From the system database: `SELECT DATABASE_NAME, SERVICE_NAME, PORT, SQL_PORT, (PORT + 2) HTTP_PORT FROM SYS_DATABASES.M_SERVICES WHERE DATABASE_NAME='<DBNAME>' and ((SERVICE_NAME='indexserver' and COORDINATOR_TYPE= 'MASTER') or (SERVICE_NAME='xsengine'))`

Remember

If your system was converted from single-container mode to multiple-container mode, the HTTP port number of the first tenant database is **always** 3<instance>08 and not the port number returned using the above queries.

Note

System privilege DATABASE ADMIN or CATALOG READ is required to read the M_SERVICES system view.

The following diagram shows an example of the connections and ports used in a multiple-container system with two tenant databases, installed on a single host. It is a new SAP HANA system that was installed in multiple-container mode, to which two tenant databases were then added.

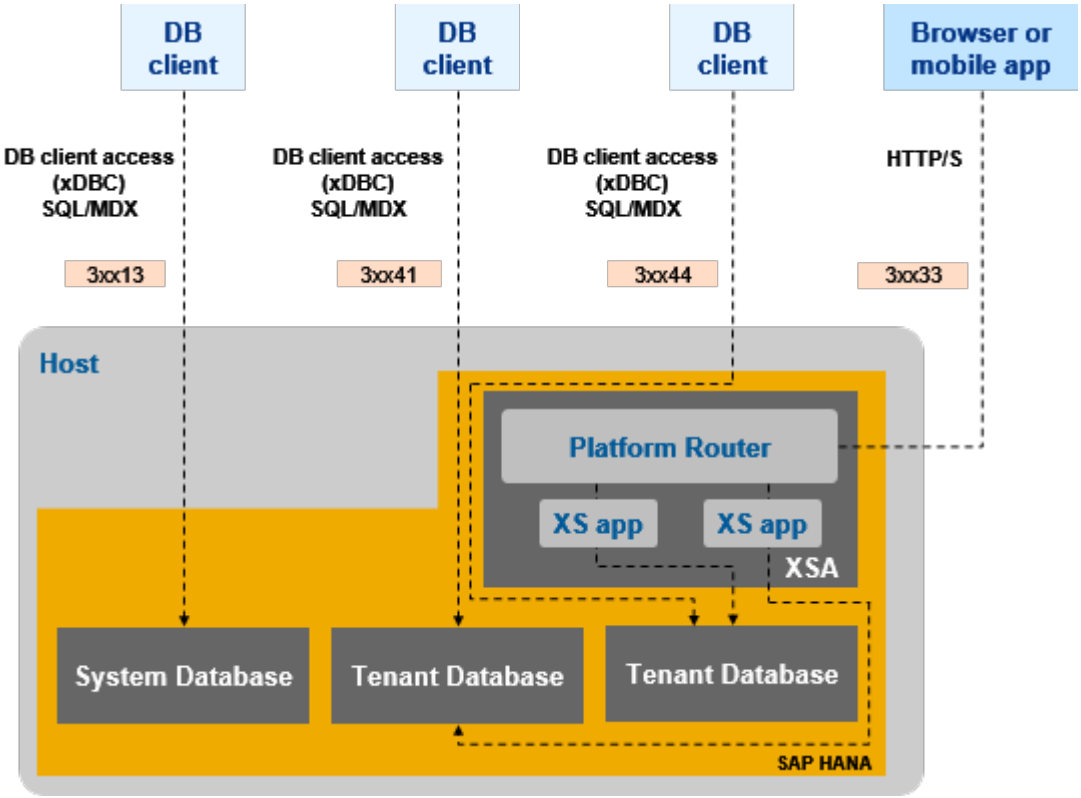


Figure 6: Connections for Multitenant Database Containers

HTTP(S) Client Access

The XS classic server allows Web-based applications to access SAP HANA via HTTP(S). The internal Web Dispatcher of the SAP HANA system manages these incoming HTTP(S) requests. To allow applications to send requests to specific databases in a multiple-container system, every tenant database needs an alias host name. Requests to the alias host name can then be forwarded to the XS server of the corresponding tenant database. Requests with the physical host name in the HTTP host header are forwarded to the XS server running on the system database.

The default HTTP ports are used in all cases, that is, 80`<instance>` (HTTP) and 43`<instance>` (HTTPS). Alias host names are mapped to internal HTTP(S) ports so that incoming requests can be routed to the correct database.

You configure HTTP(S) access to tenant databases by specifying in the `xsengine.ini` file the URLs by which each tenant database is publicly accessible. The system then automatically configures the Web Dispatcher by generating the required profile entries in the `webdispatcher.ini` configuration file. It is not necessary to specify the URL of the system database, this is done automatically.

For more information, see *Configure HTTP Access to Multitenant Database Containers* in the *SAP HANA Administration Guide*.

Related Information

Converting an SAP HANA System to Support Multitenant Database Containers [page 37]

SAP HANA Master Guide
SAP HANA Administration Guide

3 Security and Multitenant Database Containers

In general, all security-related features of the SAP HANA database (such as authentication, authorization, encryption, and auditing) apply in systems that support multitenant database containers in the same way as in systems that do not. Some restrictions and additions apply.

The following table provides an overview of standard SAP HANA security-related features in the context of multitenant database containers:

Table 2:

Security-Related Feature	Multitenant Database Containers
User and role management	Every tenant database has its own database users and roles, including a tenant database-specific superuser SYSTEM. Depending on the isolation level of the system, there may be only one operating system (OS) user (the default `<sid>`adm user), or one OS user for each tenant database, which must be created. For more information about database isolation levels, see the section *Additional Security Features in Multitenant Database Containers* below. For more information, see *Standard Users* in the *SAP HANA Security Guide* and *SYSTEM User in Multitenant Database Containers*.
Authentication and SSO	All user authentication mechanisms supported by SAP HANA are also supported in tenant databases. Whether a per-database configuration is possible depends on the authentication mechanism and the user client: • Basic authentication (with user name and password) is database specific. • For Kerberos-based authentication, a per-database configuration is not possible. Databases users in all databases must be mapped to users in the same Key Distribution Center. • For SAML-based authentication, a per-database configuration is possible for JDBC/ODBC client access. Different trust stores (containing different certificates) can be configured for individual databases. For this purpose, we recommend using certificates and certificate collections (also referred to as personal security environments or PSEs) stored in the database as opposed to the file system. • Database-specific trust stores cannot be configured for HTTP client access through SAP HANA Extended Services (SAP HANA XS). Therefore, user authentication based on SAML assertions and X.509 certificates cannot be database specific. For more information, see *SAP HANA Authentication and Single Sign-On* in the *SAP HANA Security Guide*.

Security-Related Feature	Multitenant Database Containers
Authorization	SAP HANA's standard authorization mechanisms apply to users managed directly in the tenant database with the following additions: • In the system database, the system privilege DATABASE ADMIN exists to allow system administrators to perform certain tasks on tenant databases (for example, stop a tenant database or back up a tenant database). • A cross-database authorization mechanism exists to support read-only queries between tenant databases. This is made possible through the association of a user in one tenant database with a user in another database. Cross-database access is disabled by default and must be enabled and configured by a system administrator before such user mappings can be set up.
Encryption of data communication in the network	Secure communication based on the Transport Layer Security (TLS)/Secure Sockets Layer (SSL) protocol can be configured separately for external communication between individual databases and JDBC/ODBC clients. Separate key and trust stores must be available and configured for each database. For this purpose, we recommend using certificates and certificate collections stored in the database as opposed to the file system. For communication with HTTP clients, a per-database configuration of TLS/SSL keys and certificates is also possible. For more information, see *Certificate Management in SAP HANA* and *SSL Configuration on the SAP HANA Server* in the *SAP HANA Security Guide*.
Encryption of data persistence layer	Data and log volume encryption can enabled for the system database and tenant databases individually. This ensures that anyone who can access the data and log volumes on disk using operating system commands cannot see the actual data and redo log entries. For more information, see *Data Volume Encryption* in the *SAP HANA Security Guide*.
Auditing	Actions performed in every tenant database and the system database can be audited individually. To ensure the privacy of tenant database audit trails, they are by default written to a database table that is local to the database being audited. Tenant database administrators cannot change the audit trail targets for their database. The system administrator can , but this is not recommended. If the audit trail target for tenant databases is changed to the syslog, audit entries contain a field *Database Name* so that it is possible to differentiate entries from different tenant databases. For more information, see *Auditing Configuration and Audit Policy Management* in the *SAP HANA Security Guide*.

Additional Security Features for Multitenant Database Containers

Database isolation

To maximize the security of a multiple-container system by preventing cross-tenant attacks through operating system mechanisms, it is possible to configure the system for high isolation. In high isolation mode, the processes of individual tenant databases must run under dedicated OS users belonging to dedicated OS groups, instead of all processes running under the single default OS user `<sid>`adm (low isolation). Data on the file system is protected using file and directory permissions.

Configuration change blacklist

To ensure the stability and performance of the overall system or for security reasons, it may be necessary to prevent certain system properties from being changed by tenant database administrators, for example, properties related to resource management. A configuration change blacklist (`multidb.ini`) is available for this purpose. This blacklist contains several critical properties by default. You can customize the default configuration as well as add further properties by editing the file in the SAP HANA studio.

Restricted features

Depending on how you are implementing SAP HANA, you may want to disable certain database features that provide direct access to the file system, the network, or other resources, for example import and export operations and backup functions. For this reason, these features can be explicitly disabled in tenant databases.

Related Information

SYSTEM User in Multitenant Database Containers [page 27]
Cross-Database Authorization in Multitenant Database Containers [page 28]
Database Isolation [page 12]
Prevent Changes to System Properties in Tenant Databases [page 60]
Default Blacklisted System Properties in Multitenant Database Containers [page 34]
Disable Features on a Tenant Database [page 53]
SAP HANA Security Guide (PDF)
SAP HANA Security Guide (HTML)

3.1 SYSTEM User in Multitenant Database Containers

Every database in a multiple-container system has its own set of database users, including the database superuser SYSTEM. When and how the SYSTEM user password is specified depends on whether the system was installed in multiple-container mode or converted to multiple-container mode.

A database-specific SYSTEM user exists in every database of a multiple-container system. In general, this user has the same status and privileges as the SYSTEM user in a single-container system. Only the SYSTEM user of the system database has additional privileges, namely the privileges required for managing tenant databases,

for example, creating and dropping databases, changing configuration (*.ini) files of databases, and performing database-specific data backups.

Password Specification

If you install your SAP HANA system in multiple-container mode, the SYSTEM user of the system database is created during installation. For every tenant database created in the system, the SYSTEM user is created and its password specified when the database is created.

If you convert an existing SAP HANA system to multiple-container mode, the system database and **one tenant database** are created during the conversion process. This tenant database contains all the data of the original system, including users, system configuration, and connection properties. The password of the SYSTEM user in this tenant database is the password of the SYSTEM user of the original system before it was converted. You must explicitly set the password of the SYSTEM user of the system database during conversion. For more information about how to do this, see *Convert an SAP HANA System to Support Multitenant Database Containers* in the *SAP HANA Administration Guide*.

For every subsequent tenant database created in the system, the SYSTEM user is created and its password specified when the database is created.

Related Information

Resetting the SYSTEM User Password in Multitenant Database Containers [page 87]
Converting an SAP HANA System to Support Multitenant Database Containers [page 37]
SAP HANA Administration Guide

3.2 Cross-Database Authorization in Multitenant Database Containers

Read-only queries between multitenant database containers are possible through the association of the requesting user with a remote identity on the remote database(s). Cross-database access is not enabled by default and must be configured before such user mappings can be set up.

Every tenant database in a multiple-container system is self-contained with its own isolated set of database users and isolated database catalog. However, to support for example cross-application reporting, cross-database SELECT queries are possible. This means that database objects such as tables and views can be local to one database but be read by users from other databases in the same system.

A user in one database can run a query that references objects in another database if the user is associated with a sufficiently privileged user in the remote database. This associated user is called a remote identity. This is the user who executes the query (or part of the query) in the remote database and therefore the user whose authorization is checked.

For more information about which object types on remote databases can be accessed using this mechanism and which local object types can access remote database objects, see *Cross-Database Access* in the *SAP HANA Administration Guide*.

Example

Assume that we have a multiple-container system with 2 tenant databases: DB1 and DB2.

USER2 in DB2 wants to query the table SCHEMA1.TABLE1 in DB1, for example, `SELECT * FROM DB1.SCHEMA1.TABLE1`.

This can be achieved as follows:

1. The administrator of DB1 creates a user in DB1 with a remote identity in DB2:

   ```
   CREATE USER USER1 WITH REMOTE IDENTITY USER2 AT DATABASE DB2
   ```

2. The administrator of DB1 grants user USER1 the privileges required to read the table SCHEMA1.TABLE1:

   ```
   GRANT SELECT ON SCHEMA1.TABLE1 TO USER1 [WITH GRANT OPTION]
   ```

Now, USER2 in DB2 can select from SCHEMA1.TABLE1 in DB1.

For more information about the syntax notation, see *CREATE USER* in the *SAP HANA SQL and System Views Reference*.

Things to Note About Remote Identities

- A user can be the remote identity for only one user in another database.
- An existing user can be assigned a remote identity using the ALTER USER statement.
- The association between a user and a remote identity is unidirectional. In the above example, USER2 can access SCHEMA1.TABLE1 in DB1 as USER1, but USER1 cannot access objects in DB2 as USER2.
- Only the SELECT privileges of the user in the remote database are considered during a cross-database query. Any other privileges the remote user may have are ignored.
- Before users with remote identities can be created, an administrator must enable cross-database access for the system in the system database and specify which databases can communicate with one another. For more information, see *Enable and Configure Cross-Database Access* in the *SAP HANA Administration Guide*.
  Users receive a `Not authorized` error if they attempt a cross-database operation that is not supported by the current configuration.

System Views for Monitoring Cross-Database Authorization

The following system views contain information about cross-database authorization in a tenant database:

- USERS (SYS)
  The column REMOTE_USER indicates whether or not a particular user in the local database has remote identities in other databases.

- REMOTE_USERS (SYS)
  This system view shows which local users can be used by users on other databases for cross-database query execution.

> **Note**
>
> The system views EFFECTIVE_PRIVILEGES and ACCESSIBLE_VIEWS **do not** include privileges that a user has through a remote identity.

Related Information

Cross-Database Access [page 11]
Enable and Configure Cross-Database Access [page 54]
SAP HANA Administration Guide
SAP HANA SQL and System Views Reference

3.3 Data and Log Volume Encryption in Multitenant Database Containers

Data volume encryption and redo log encryption can be enabled individually for tenant databases in a multiple-container system. All databases have their own root keys.

Ideally, you enable encryption immediately after installation or upgrade of SAP HANA. This also applies to systems installed in multiple-container mode. Any subsequently created tenant databases will then automatically have encryption enabled. If a particular tenant database does not require encryption, the tenant database administrator can switch it off independently of the system.

> **Note**
>
> If encryption is enabled or disabled in the system database after creation of the tenant database, the tenant database will not inherit this setting, even if the setting has never been made individually in the tenant database.

If encryption is not enabled after system installation, you can enable for individual tenant databases by making the setting in the relevant tenant database.

> **Caution**
>
> If you enable data volume encryption after a tenant database has been created and is already in operation, only the pages in use within the data volumes are encrypted. Pages in the data volumes that are not in use may still contain old content and will only be overwritten and encrypted over time. This means that data in data volumes will only be fully encrypted after some delay. Furthermore, if you enable redo log encryption, only future redo log entries are encrypted. Previous redo log entries remain unencrypted.

To attain complete protection immediately, the overall process is:

1. Perform a data backup.
2. Drop the tenant database.
3. Clean the disk space.
4. Create the tenant database again.
5. Enable data volume encryption and redo log encryption.
6. Perform a data recovery.

3.4 Restricted Features in Multitenant Database Containers

To safeguard and/or customize your system, certain features of the SAP HANA database can be disabled in tenant databases.

Some features of the SAP HANA database are not required or desirable in certain environments, in particular features that provide direct access to the file system, the network, or other resources. The table below lists those features that you can explicitly disable in tenant databases.

The system view M_CUSTOMIZABLE_FUNCTIONALITIES lists all features that can be disabled and their status. This view exists in both the SYS schema of every database, where it contains database-specific information, and in the SYS_DATABASES schema of the system database, where it contains information about the enablement of features in all databases. For more information, see *M_CUSTOMIZABLE_FUNCTIONALITIES* in the *SAP HANA SQL and Systems View Reference*.

For more information about how to disable features, see *Disable Restricted Features on a Tenant Database* in the *SAP HANA Administration Guide*.

> **Note**
>
> Features are hierarchically structured. If you disable a feature with sub-features, these are also disabled.

Table 3:

Feature	Feature Description	Why Disable?
AFL	Access to Application Function Libraries (AFL) for business logic in native C ++	Feature not required in all deployment scenarios
BACKUP	Backup operations	File system access not wanted
IMPORTEXPORT	Import and export operations	File system access not wanted
IMPORTEXPORT.IMPORT	Import operations	File system read access not wanted
IMPORTEXPORT.EXPORT	Export operations	File system write access not wanted
IMPORTEXPORT.IGNORE_PATH_RESTRICT	Ignoring of path restrictions for import and export	File system access not wanted, safeguard against directory traversal attacks

Feature	Feature Description	Why Disable?
BUILTINPROCEDURE	Execution of procedures associated with critical and/or optional functions	--
BUILTINPROCEDURE.MANAGEMENT_CONSOLE_PROC	Access to the built-in SAP HANA management console (`hdbcons`)	Safeguard against leakage of SAP HANA process information
BUILTINPROCEDURE.GEM	Procedure to use the graph engine	Feature not required in all deployment scenarios
BUILTINPROCEDURE.KERNELCALL	Access to rowstore internal maintenance features	Safeguard against leakage of SAP HANA process information
BUILTINPROCEDURE.TREXVIADBSL	Operation of an SAP Business ByDesign system	Feature not required in all deployment scenarios
BUILTINPROCEDURE.COMPRESS_FILE	Compression of trace files before they are transferred	Feature not required in all deployment scenarios
BUILTINPROCEDURE.GET_FULL_SYSTEM_INFO_DUMP	Triggering of complete information dump of the entire system	Feature not required in all deployment scenarios
BUILTINPROCEDURE.DSO	Creation of and access to DataStore Objects (DSOs) for SAP Business Warehouse (BW) powered by SAP HANA	Feature not required in all deployment scenarios
BUILTINPROCEDURE.STATISTICSSERVER_CONFIGCHECKPROC	Validation of the statisticsserver configuration and its e-mail notification capability	Feature not required in all deployment scenarios
BUILTINPROCEDURE.BW_PRECHECK_RELEASE_LOCK	Operation of an SAP BW powered by SAP HANA system	Feature not required in all deployment scenarios
BUILTINPROCEDURE.BW_PRECHECK_ACQUIRE_LOCK_WITH_TYPE		
BUILTINPROCEDURE.BW_PRECHECK_ACQUIRE_LOCK		
BUILTINPROCEDURE.BW_CONVERT_CLASSIC_TO_IMO_CUBE		
BUILTINPROCEDURE.BW_F_FACT_TABLE_COMPRESSION		
BUILTINPROCEDURE.UPDATE_LANDSCAPE_CONFIGURATION	Changes to system landscape and the available services in a system	Feature not required in all deployment scenarios
BUILTINPROCEDURE.REORG_GENERATE	Data redistribution operations	Feature not required in all deployment scenarios
BUILTINPROCEDURE.REORG_EXECUTE		
BUILTINPROCEDURE.REORG_CLEAR_LOGS		
ALTERSYSTEM	Execution of the statement ALTER SYSTEM RECONFIGURE SERVICE, which re-reads the service configuration	Feature not required in all deployment scenarios
ALTERSYSTEM.RECONFIGURE_SERVICE		

Feature	Feature Description	Why Disable?
SMARTDATAACCESS	Federated access to other database systems through virtual tables	Feature not required in all deployment scenarios
DXC	Data acquisition and consumption of data models from the SAP Business Suite	Feature not required in all deployment scenarios
DYNAMIC_TIERING	SAP HANA Dynamic Tiering operations	Feature not required in all deployment scenarios
DYNAMIC_TIERING.CREATE_EXTENDED_STORAGE	Creation of extended storage	
DYNAMIC_TIERING.DROP_EXTENDED_STORAGE	Deletion of extended storage	
DYNAMIC_TIERING.ALTER_EXTENDED_STORAGE	Changes to extended storage	
DYNAMIC_TIERING.ALTER_TABLE_TYPE	Conversion of a regular database table to an extended table or the reverse	
DYNAMIC_TIERING.BULK_INSERT_OPTIMIZATION	Bulk insert optimization that executes large inserts into extended tables using a load statement	
DYNAMIC_TIERING.QUERY_PLAN_RELOCATION	Query relocation operation that moves data from SAP HANA and SAP HANA Dynamic Tiering for optimal query performance	
ACCELERATOR_FOR_ASE	SAP HANA Accelerator for SAP ASE operations	
EPMPLANNING	SAP HANA Enterprise Performance Management planning feature	Feature not required in all deployment scenarios
PLANNINGENGINE	Planning engine features	Feature not required in all deployment scenarios
RINTEGRATION	R language	Feature not required in all deployment scenarios
BOE	SAP BusinessObjects Explorer API	Feature not required in all deployment scenarios

Related Information

Disable Features on a Tenant Database [page 53]

SAP HANA Administration Guide

SAP HANA SQL and System Views Reference

3.5 Default Blacklisted System Properties in Multitenant Database Containers

In systems that support multitenant database containers, there is configuration change blacklist `multidb.ini`, which is delivered with a default configuration.

The table below lists the system properties that are included in the `multidb.ini` file by default. This means that tenant database administrators cannot change these properties. System administrators can still change these properties in the system database in all layers.

You can customize the default configuration change blacklist by changing existing entries in the `multidb.ini` file and adding new ones. For more information, see *Prevent Changes to Specific System Properties in Tenant Databases* in the *SAP HANA Administration Guide*.

Table 4:

File/Section	Properties	Description
`auditing configuration`	• `default_audit_trail_type` • `emergency_audit_trail_type` • `alert_audit_trail_type` • `critical_audit_trail_type`	Prevents configuration of audit trail targets
`indexserver.ini/authentication`	`SapLogonTicketTrustStore`	Prevents configuration of the trust store for user authentication with logon/assertion tickets
`communication`	*	Prevents configuration of default key and trust stores, as well as other critical communication settings
`global.ini/customizable_functionalities`	*	Prevents disabling of restricted features
`global.ini/extended_storage`	*	Prevents configuration of extended storage (SAP HANA dynamic tiering option)
`global.ini/persistence`	• `basepath_datavolumes_es` • `basepath_logvolumes_es` • `basepath_databackup_es` • `basepath_logbackup_es`	
`multidb.ini/readonly_parameters`	*	Prevents configuration of the `multidb.ini` file itself

File/Section	Properties	Description
memorymanager	• allocationlimit • minallocationlimit • global_allocation_limit • async_free_threshold • async_free_target	Prevents configuration of memory allocation parameters
execution	max_concurrency	Prevents configuration of threading and parallelization parameters
session	• maximum_connections • maximum_external_connections	
sql	sql_executors	

Related Information

Prevent Changes to System Properties in Tenant Databases [page 60]
SAP HANA Administration Guide

4 Creating and Configuring Tenant Databases

During the installation of a multiple-container system, only the system database is initially created. You create and configure tenant databases afterward. If your system was updated or installed in single-container mode, you must convert it to multiple-container mode before setting up tenant databases.

As a system administrator, you create tenant databases from the system database. You can then configure the new databases as required:

- Increase the database isolation level
- Disable certain features that are not required in tenant databases (for example, backup operations)
- Enable and configure cross-database access if read-only queries between tenant databases is required
- Edit the configuration change blacklist so that critical system properties cannot be changed by tenant database administrators
- Configure the SAP Web Dispatcher if tenant databases will be accessed by HTTP clients via the SAP HANA XS classic server

> **Note**
>
> Administration of tenant databases is possible using the SAP HANA cockpit. However, command-line tools are required for some tasks.

Related Information

Converting an SAP HANA System to Support Multitenant Database Containers [page 37]
Increase the System Isolation Level [page 46]
Create a Tenant Database [page 50]
Disable Features on a Tenant Database [page 53]
Enable and Configure Cross-Database Access [page 54]
Prevent Changes to System Properties in Tenant Databases [page 60]
Configure HTTP(S) Access to Multitenant Database Containers via SAP HANA XS Classic [page 62]
Administration of Multitenant Database Containers [page 14]

4.1 Converting an SAP HANA System to Support Multitenant Database Containers

You can convert an SAP HANA system to support multitenant database containers using the SAP HANA database lifecycle manager (HDBLCM) resident program. Converting an SAP HANA system to a multiple-container system is permanent and cannot be reversed.

If your system was installed in single-container mode, you can still implement multitenant database containers by converting the system to a multiple-container system. During the conversion process, the system database and one tenant database are created. The tenant database contains all the data of the original system, including users, system configuration, connection properties (port configuration), and system license. However, it does **not** contain the backup history.

After conversion, you can create and configure further tenant databases as needed.

Related Information

Convert to Multitenant Database Containers Using the Graphical User Interface [page 37]
Convert to Multitenant Database Containers Using the Command-Line Interface [page 39]
Convert to Multitenant Database Containers Using the Web User Interface [page 41]

4.1.1 Convert to Multitenant Database Containers Using the Graphical User Interface

You can convert an SAP HANA system to support multitenant database containers using the SAP HANA database lifecycle manager (HDBLCM) resident program in the graphical user interface. Converting an SAP HANA system to a multiple-container system is permanent and cannot be reversed.

Prerequisites

- The statistics server is **not** running as a separate server process (`statisticsserver`), but instead as an embedded service in the master index server. If this is not the case, migrate the statistics server to the embedded statistics service as described in SAP Note 1917938.
- The SAP HANA system has been installed with its server software on a shared file system (export options `rw`, `no_root_squash`).
- The host has access to the installation directories `<sapmnt>` and `<sapmnt>/<SID>`.
- The SAP HANA system has been installed with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.
- You are logged on as root user or as the system administrator user `<sid>`adm.

- The difference between the system time set on the installation host and the additional host is not greater than 180 seconds.
- The operating system administrator (`<SID>`adm) user may exist on the additional host. Make sure that you have the password of the existing `<SID>`adm user, and that the user attributes and group assignments are correct. The SAP HANA database lifecycle manager (HDBLCM) resident program will not modify the properties of any existing user or group.

Procedure

1. Change to the SAP HANA resident HDBLCM directory:

   ```
   cd <sapmnt>/<SID>/hdblcm
   ```

   By default, `<sapmnt>` is `/hana/shared`.
2. Start the SAP HANA database lifecycle manager interactively in the graphical user interface:

   ```
   ./hdblcmgui
   ```

   The SAP HANA database lifecycle manager graphical user interface appears.
3. Select *Convert to Multitenant Database Containers* from the activity options. Then select *Next*.
4. Provide the password of the `<sid>adm` user and the `SYSTEM` user of `SYSTEMDB`, then select *Next*.
5. Review the summary, and select *Run* to finalize the configuration.

Results

Your SAP HANA system is a multiple-container system with one system database and one tenant database, both of which are running. You can verify this by querying the public view M_DATABASES from the system database. The result will look like this:

```
| DATABASE_NAME   |DESCRIPTION                        | ACTIVE_STATUS |
|-----------------|-----------------------------------|---------------|
| SYSTEMDB        | SystemDB-<SID>-<INSTANCE>         | YES           |
| <SID>           | SingleDB-<SID>-<INSTANCE>         | YES           |
```

Note the following about the tenant database:

- It contains all the data (including users, configuration, and connection properties) of the original system (but not the original backup history).
- Configuration files that are tenant-specific (e.g. indexserver.ini, xsengine.ini, etc.) are now stored at the following location: /usr/sap/`<SID>`/SYS/global/hdb/custom/config/DB_`<database_name>`.
- Its trace files are now stored at the following location: `/usr/sap/<SID>/HDB<instance>/<host>/trace/DB_<database_name>`.

> **Note**
>
> Any trace files that were in the trace directory before the system was converted are not moved.

Next Steps

- Create and configure any additionally required tenant databases. For more information, see *Create a Tenant Database*.

  > **Note**
  >
  > If you configured the properties of the index server, script server, or xsengine server in your original system, these settings initially apply to **all** new tenant databases. You must explicitly configure tenant database if required. For more information, see *System Properties in Multiple-Container Systems* in the *SAP HANA Administration Guide*.

- If HTTP access via the SAP HANA XS classic server is required, update the configuration of the Web Dispatcher. For more information, see *Configure HTTP Access to Multitenant Database Containers* in the *SAP HANA Administration Guide*.

Related Information

SAP Note 1917938
Create a Tenant Database [page 50]
Configuration Parameters in Multiple-Container Systems [page 15]
Configure HTTP(S) Access to Multitenant Database Containers via SAP HANA XS Classic [page 62]
SAP HANA Server Installation and Update Guide
SAP HANA Administration Guide

4.1.2 Convert to Multitenant Database Containers Using the Command-Line Interface

You can convert an SAP HANA system to support multitenant database containers using the SAP HANA database lifecycle manager (HDBLCM) resident program in the command-line interface. Converting an SAP HANA system to a multiple-container system is permanent and cannot be reversed.

Prerequisites

- The statistics server is **not** running as a separate server process (`statisticsserver`), but instead as an embedded service in the master index server. If this is not the case, migrate the statistics server to the embedded statistics service as described in SAP Note 1917938.
- The SAP HANA system has been installed with its server software on a shared file system (export options `rw`, `no_root_squash`).
- The host has access to the installation directories `<sapmnt>` and `<sapmnt>/<SID>`.

- The SAP HANA system has been installed with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.
- You are logged on as root user or as the system administrator user `<sid>`adm.
- The difference between the system time set on the installation host and the additional host is not greater than 180 seconds.
- The operating system administrator (`<SID>`adm) user may exist on the additional host. Make sure that you have the password of the existing `<SID>`adm user, and that the user attributes and group assignments are correct. The SAP HANA database lifecycle manager (HDBLCM) resident program will not modify the properties of any existing user or group.

Procedure

1. Change to the SAP HANA resident HDBLCM directory:

   ```
   cd <sapmnt>/<SID>/hdblcm
   ```

   By default, `<sapmnt>` is `/hana/shared`.
2. Start the SAP HANA database lifecycle manager interactively in the command line:

   ```
   ./hdblcm --action=convert_to_multidb
   ```
3. Provide the password of the `<sid>adm` user and `SYSTEM` user of `SYSTEMDB` user.
4. Review the summary, and select *y* to finalize the configuration.

Results

Your SAP HANA system is a multiple-container system with one system database and one tenant database, both of which are running. You can verify this by querying the public view M_DATABASES from the system database. The result will look like this:

```
| DATABASE_NAME   |DESCRIPTION                        | ACTIVE_STATUS |
|-----------------|-----------------------------------|---------------|
| SYSTEMDB        | SystemDB-<SID>-<INSTANCE>         | YES           |
| <SID>           | SingleDB-<SID>-<INSTANCE>         | YES           |
```

Note the following about the tenant database:

- It contains all the data (including users, configuration, and connection properties) of the original system (but not the original backup history).
- Configuration files that are tenant-specific (e.g. indexserver.ini, xsengine.ini, etc.) are now stored at the following location: /usr/sap/`<SID>`/SYS/global/hdb/custom/config/DB_`<database_name>`.
- Its trace files are now stored at the following location: `/usr/sap/<SID>/HDB<instance>/<host>/trace/DB_<database_name>`.

> **Note**
>
> Any trace files that were in the trace directory before the system was converted are not moved.

Next Steps

- Create and configure any additionally required tenant databases. For more information, see *Create a Tenant Database*.

  > **Note**
  >
  > If you configured the properties of the index server, script server, or xsengine server in your original system, these settings initially apply to **all** new tenant databases. You must explicitly configure tenant database if required. For more information, see *System Properties in Multiple-Container Systems* in the *SAP HANA Administration Guide*.

- If HTTP access via the SAP HANA XS classic server is required, update the configuration of the Web Dispatcher. For more information, see *Configure HTTP Access to Multitenant Database Containers* in the *SAP HANA Administration Guide*.

Related Information

SAP Note 1917938
Create a Tenant Database [page 50]
Configuration Parameters in Multiple-Container Systems [page 15]
Configure HTTP(S) Access to Multitenant Database Containers via SAP HANA XS Classic [page 62]
SAP HANA Server Installation and Update Guide
SAP HANA Administration Guide
import_content [page 45]
nostart [page 46]
nostart_tenant_db [page 46]

4.1.3 Convert to Multitenant Database Containers Using the Web User Interface

You can convert an SAP HANA system to support multitenant database containers using the SAP HANA database lifecycle manager Web user interface. Converting an SAP HANA system to a multiple-container system is permanent and cannot be reversed.

Prerequisites

- The statistics server is **not** running as a separate server process (`statisticsserver`), but instead as an embedded service in the master index server. If this is not the case, migrate the statistics server to the embedded statistics service as described in SAP Note 1917938.

- The difference between the system time set on the installation host and the additional host is not greater than 180 seconds.
- The SAP HANA system has been installed with its server software on a shared file system (export options `rw`, `no_root_squash`).
- The host has access to the installation directories `<sapmnt>` and `<sapmnt>/<SID>`.
- The SAP HANA system has been installed or updated with the SAP HANA database lifecycle manager (HDBLCM).
- The SAP HANA database server is up and running.

You should verify that the following prerequisites are fulfilled before trying to access the SAP HANA database lifecycle manager from a Web browser.

- The communication port 1129 is open.
  Port 1129 is required for the SSL communication with the SAP Host Agent in a standalone browser via HTTPS.
- The following Web browser requirements are fulfilled:
  - Microsoft Windows
    - Internet Explorer - Version 9 or higher
      If you are running Internet Explorer version 9, make sure that your browser is not running in compatibility mode with your SAP HANA host. You can check this in your browser by choosing *Tools* > *Compatibility View Settings*.
    - Microsoft Edge
    - Mozilla Firefox - Latest version and Extended Support Release
    - Google Chrome - Latest version
  - SUSE Linux - Mozilla Firefox with XULRunner 10.0.4 ESR
  - Mac OS - Safari 5.1 or higher

  Note

  For more information about supported Web browsers for the SAP HANA database lifecycle manager Web interface, see the browser support for `sap.m` library in the *SAPUI5 Developer Guide* in Related Information.
- You are logged on as the system administrator user `<sid>`adm.

You should verify that the following additional prerequisites are fulfilled before trying to access the SAP HANA database lifecycle manager from the SAP HANA studio.

- The SAP HANA studio revision is 120 or higher.
- For Linux:
  - The system property `org.eclipse.swt.browser.XULRunnerPath` should be set in `hdbstudio.ini` to point to the path of XULRunner, for example:
    `-Dorg.eclipse.swt.browser.XULRunnerPath=<path to xulrunner>`.
    This `hdbstudio.ini` file is located in the same folder as the executable that is used to start the SAP HANA studio. For Linux, the default location is `hana/shared/<SID>/hdbstudio`.

Procedure

1. Access the SAP HANA HDBLCM Web user interface.

Option	Description
Web Browser	Enter the SAP HANA database lifecycle manager (HDBLCM) URL in an HTML5-enabled browser: https://`<hostname>`:1129/lmsl/HDBLCM/`<SID>`/index.html **Note** The URL is case sensitive. Make sure you enter upper and lower case letters correctly.
SAP HANA Studio	1. Start the SAP HANA studio. 2. In the SAP HANA studio, add the SAP HANA system. 3. Open the context menu (right-mouse click) in the *Systems* view, and select *Add System*. For more information about adding a system, see *Add an SAP HANA System* in the *SAP HANA Administration Guide* in Related Information. 4. In the SAP HANA studio, log on to the system. 5. From the context menu of the selected system, select *Lifecycle Management* > *Platform Lifecycle Management* > *SAP HANA Platform Lifecycle Management*.
SAP HANA Cockpit	1. Enter the SAP HANA cockpit URL in your browser. **Note** FQDN = fully qualified domain name 2. The *SAP HANA Platform Lifecycle Management* tiles are visible on the homepage of the SAP HANA cockpit. If they are not, you can add them from the *SAP HANA Platform Lifecycle Management* tile catalog. For more information, see *Customizing the Homepage of SAP HANA Cockpit*.

2. Select the *Convert to Multitenant Database Containers* tile.
3. Optional: Modify the following parameters in the *Advanced Parameters Configuration* dialog. To access the *Advanced Parameters Configuration* dialog, click on the gear icon in the footer bar of the SAP HANA HDBLCM Web user interface.

Option	Description
Import Delivery Units In The System Database	Import Delivery Units In The System Database
Do Not Start Instance After Reconfiguration	Do Not Start Instance After Reconfiguration
Do Not Start Tenant Database After Reconfiguration	Do Not Start Tenant Database After Reconfiguration
Timeouts	Sets customized timeouts (`start_instance`, `stop_instance`).

4. Provide the password of the `<sid>adm` user and `SYSTEM` user of `SYSTEMDB` user, then select *Next*.
5. Review the summary, and select *Run* to finalize the configuration.

Results

Your SAP HANA system is a multiple-container system with one system database and one tenant database, both of which are running. You can verify this by querying the public view M_DATABASES from the system database. The result will look like this:

```
| DATABASE_NAME   |DESCRIPTION                        | ACTIVE_STATUS |
|-----------------|-----------------------------------|---------------|
| SYSTEMDB        | SystemDB-<SID>-<INSTANCE>         | YES           |
| <SID>           | SingleDB-<SID>-<INSTANCE>         | YES           |
```

Note the following about the tenant database:

- It contains all the data (including users, configuration, and connection properties) of the original system (but not the original backup history).
- Configuration files that are tenant-specific (e.g. indexserver.ini, xsengine.ini, etc.) are now stored at the following location: /usr/sap/`<SID>`/SYS/global/hdb/custom/config/DB_`<database_name>`.
- Its trace files are now stored at the following location: `/usr/sap/<SID>/HDB<instance>/<host>/trace/DB_<database_name>`.

> **Note**
>
> Any trace files that were in the trace directory before the system was converted are not moved.

Next Steps

- Create and configure any additionally required tenant databases. For more information, see *Create a Tenant Database*.

  > **Note**
  >
  > If you configured the properties of the index server, script server, or xsengine server in your original system, these settings initially apply to **all** new tenant databases. You must explicitly configure tenant database if required. For more information, see *System Properties in Multiple-Container Systems* in the *SAP HANA Administration Guide*.

- If HTTP access via the SAP HANA XS classic server is required, update the configuration of the Web Dispatcher. For more information, see *Configure HTTP Access to Multitenant Database Containers* in the *SAP HANA Administration Guide*.

Related Information

SAP Note 1917938
Create a Tenant Database [page 50]
Configuration Parameters in Multiple-Container Systems [page 15]
Configure HTTP(S) Access to Multitenant Database Containers via SAP HANA XS Classic [page 62]

SAP HANA Server Installation and Update Guide
SAP HANA Administration Guide

4.1.4 Parameter Reference: Converting an SAP HANA System to Support Multitenant Database Containers

Parameters can be specified when converting an SAP HANA system to multitenant database containers in order to customize the configuration task.

The SAP HANA database lifecycle manager convert to multidb action also supports the following parameters:

- `batch`
- `configfile`
- `dump_configfile_template`
- `help`
- `list_systems`
- `read_password_from_stdin`
- `version`

For more information about these parameters, see the *SAP HANA Server Installation and Update Guide*

For a complete list of the parameters, call the help of the convert to multidb task with the following command:

```
./hdblcm --action=convert_to_multidb --help
```

Related Information

SAP HANA Server Installation and Update Guide

4.1.4.1 `import_content`

Imports delivery units.

Syntax

In the command line, the following syntax is used:

```
--import_content[=off]
```

Remarks

The default for this parameter is `--import_content`.

4.1.4.2 `nostart`

Prevents the SAP HANA system from being started.

Syntax

In the command line, the following syntax is used:

```
--nostart
```

4.1.4.3 `nostart_tenant_db`

Prevents the SAP HANA tenant databases from being started.

Syntax

In the command line, the following syntax is used:

```
--nostart_tenant_db
```

4.2 Increase the System Isolation Level

You can increase the isolation level of an existing multiple-container system from low (default) to high. With high isolation, the processes of individual tenant databases run under dedicated operating system (OS) users belonging to dedicated (OS) groups and internal communication is secured.

Prerequisites

- You have root access to the SAP HANA system.
- You are logged on to the system database in the SAP HANA cockpit.
- You have the system privilege `DATABASE ADMIN`.
- Internal SAP HANA communication has been appropriately configured for TLS/SSL. `[communication] ssl` in the `global.ini` file must have the value **`false`** (default) or **`systemPKI`**.

  > ⚠ Caution
  >
  > The value of the The property `[communication] ssl` property must **not** be **`true`**. This will be the case if you are using a manually configured public key infrastructure (PKI) to secure internal communication between hosts. You can switch to system PKI by changing setting the parameter to **`systemPKI`**.

  For more information, see *Secure Internal Communication* in the *SAP HANA Security Guide*.

- If the system is running in an SAP HANA system replication configuration, the system PKI SSFS data file and key file have been copied from the primary system to the same location on the secondary system(s):
  - `$DIR_INSTANCE/../global/security/rsecssfs/data/SSFS_<SID>.DAT`
  - `$DIR_INSTANCE/../global/security/rsecssfs/key/SSFS_<SID>.KEY`

Procedure

1. For every tenant database, create a dedicated OS user and group:
   a. As root user, log on to the server on which the name server of the system database is running.
   b. Create new groups for every tenant database:

      ```
      groupadd <groupname>
      ```

   c. Create new users for every tenant database, specifying `sapsys` as the primary group:

      ```
      useradd -g sapsys <username>
      ```

   d. Add every new user to the `sidshm` group and their own group as secondary groups:

      ```
      usermod -G <sid>shm,<usergroup> <username>
      ```

   > **Note**
   >
   > If the system is distributed across multiple hosts, you must create identical users and groups on every host. Users and groups must have the same names and IDs on all hosts.

2. Stop all tenant databases in the system.

   You do this in the system database in one of the following ways:

   - Executing the SQL statement `ALTER SYSTEM STOP DATABASE <databasename>`
   - On the *Manage Databases* page of the SAP HANA cockpit

3. Configure the system for high isolation.

   As the operating system user `<sid>`adm, log on to the server on which the master index server is running and run the following command:

   ```
   python /usr/sap/<SID>/HDB<instance>/exe/python_support/convertMDC.py --
   change=databaseIsolation --isolation=high
   ```

   This command runs the following actions:

   - Stops the system
   - Changes the value of the `[multidb] database_isolation` property in the `global.ini` file to `high`
   - Starts the system

4. Assign every database to their respective OS user and group.

   In the system database, execute the SQL statement `ALTER DATABASE <databasename> OS USER '<username>' OS GROUP '<groupname>'`

5. Start all tenant databases.

   You do this in the system database in one of the following ways:

   - Executing the SQL statement `ALTER SYSTEM START DATABASE <database_name>`

- On the *Manage Databases* app of the SAP HANA cockpit

Results

The system is now running in high isolation mode. As a result:

- The processes of individual tenant databases run under dedicated OS users belonging to dedicated OS groups and the processes of the system database run under the `<sid>`adm user.
- Internal database communication is authenticated using X.509 client certificates. Depending on how SSL for internal communication is configured, data communication within databases may also be encrypted. For more information, see *Securing Internal Communication* in the *SAP HANA Security Guide*.
- Operations that require operating system access are restricted to users with the correct permissions. For more information, see *File and Directory Permissions with High Isolation*.
- New tenant databases can only be created if a dedicated OS user and group exist.

Related Information

Database Isolation [page 12]
Stop and Start a Tenant Database [page 79]
Create a Tenant Database [page 50]
SAP HANA Security Guide

4.2.1 File and Directory Permissions with High Isolation

In a multiple-container system configured for high isolation, database-specific data on the file system is protected using standard file and directory permissions.

The following table shows who has access to which data on the file system:

Table 5:

File and Directory Permissions	Tenant OS User in Tenant OS Group	`<sid>`adm User
Read, write, and execute permissions (0770) on database-specific directories containing: • Data volumes • Log volumes • Log mirror volumes • Backups	Yes	No

File and Directory Permissions	Tenant OS User in Tenant OS Group	<sid>adm User
Read, write, and execute permissions (0770) on database-specific directories containing: • Configuration (*ini) files • Trace files	Yes	Yes
Read and write permissions (0666) on files in database-specific directory containing: • Configuration (*ini) files • Trace files	Yes	Yes
Read permission (644) on files in directory containing system configuration files	Yes	Yes
Read and write permissions (600) on files in trace directory of the system database	No	Yes

4.3 Decrease the System Isolation Level

If you configured a multiple-container system for high isolation during installation or later, you can decrease it back to the default low level if necessary. With low isolation, the processes of all databases run under the default operating system (OS) user <sid>adm.

Prerequisites

- You have root access to the SAP HANA system.
- You are logged on to the system database in the SAP HANA cockpit.
- You have the system privilege DATABASE ADMIN.

Procedure

1. Stop all tenant databases in the system.

   You do this in the system database in one of the following ways:

   - Executing the SQL statement `ALTER SYSTEM STOP DATABASE <database_name>`
   - On the *Manage Databases* page of the SAP HANA cockpit
2. Configure the system for low isolation.

As the operating system user `<sid>`adm, log on to the server on which the master index server is running and run the following command:

```
python /usr/sap/<SID>/HDB<instance>/exe/python_support/convertMDC.py --
change=databaseIsolation --isolation=low
```

This command runs the following actions:

- Stops the system
- Changes the value of the `[multidb] database_isolation` property in the `global.ini` file to `low`
- Starts the system

3. Clear the assignment of OS users and groups to tenant databases.

   In the system database, execute the SQL statement `ALTER DATABASE <database_name> OS USER '' OS GROUP ''` for every tenant database.

4. Start all tenant databases.

   You do this in the system database in one of the following ways:

   - Executing the SQL statement `ALTER SYSTEM START DATABASE <databasename>`
   - On the *Manage Databases* page of the SAP HANA cockpit

Results

The system is now running in low isolation mode again.

- The processes of all databases run under `<sid>`adm.
- Internal database communication is not authenticated.

Related Information

Stop and Start a Tenant Database [page 79]

4.4 Create a Tenant Database

You create tenant databases after installation of a multiple-container system, after conversion from a single-container system to a multiple-container system, or anytime a new database is needed. You create tenant databases from the system database using *Manage Databases* in the SAP HANA cockpit.

Prerequisites

- You have the system privilege `DATABASE ADMIN`.

- If the system is configured for high isolation, the operating system (OS) user and group required for the new tenant database already exist. For more information, see *Database Isolation* in the *SAP HANA Administration Guide*.

Procedure

1. Open *Manage Databases* by drilling down from the *Overall System Status* tile on the *Overview* page of the system database in the SAP HANA cockpit.
2. In the footer toolbar, open the overflow menu (by clicking the three dots) and choose *Create Tenant Database*.
   The *Create Tenant Database* page opens.
3. Enter the name of the new database and the password of the SYSTEM user.

   > **Note**
   >
   > The password must initially comply with the password policy configured in the system database. Once the database is created, you can change the password policy for the tenant database if you want.

4. Optional: Specify the OS user and group of the tenant database.

   If the system in which you are creating the tenant database is configured for high isolation, the processes of individual tenant databases **must** run under dedicated OS users in dedicated OS groups.

   > **Note**
   >
   > If the system is configured for low isolation (default), all tenant database processes run under the default OS user `<sid>`adm.

5. Optional: Prevent the database from being started immediately after creation.

   By default, the tenant database will be started immediately after creation. If you don't want this to happen, open the *Advanced Settings* section and deselect the *Start Automatically* option.

   > **Example**
   >
   > You want to configure the new database before starting it to avoid having to restart.

6. Optional: Specify the host on which the database is to be created.

   If the system is distributed across multiple hosts, you can specify on which host you want the master index server to start. You do this in the *Advanced Settings* section by selecting the host for the default service. If you don't select a host, load-balancing algorithms will determine optimal host placement.
7. Optional: Specify the number of the internal communication port of the master index server.

   You do this in the *Advanced Settings* section by entering the port number for the default service. If you don't enter a port, it is assigned automatically based on port number availability. For more information about port number assignment, see *Connections for Multitenant Database Containers* in the *SAP HANA Master Guide*.
8. Optional: Add any additionally required services.
   a. In the *Advanced Settings* section, choose *Add Service*.

b. Select the service you want to add.
c. Optional: Select the host and enter the port number of the new service.

   If you don't select a host or enter a port number, they will be automatically determined.

9. Click *Create Tenant Database*.

   The system starts creating the database. This may take a few moments to complete.

   Technically, the creation process runs in the background as follows:

   - The database is assigned a unique system local ID.
   - If you did not specify host information, load-balancing algorithms determine optimal host placement.
   - If you did not specify the number of the internal communication port, it is assigned automatically based on port number availability.
   - The necessary data and log volumes are created on the affected hosts.
   - The new database is entered in the M_DATABASES system view of the system database.
   - The `daemon.ini` file is updated and the daemon process is triggered to start the indexserver service and any additionally added services on each configured host.
   - The specified password is set for the user SYSTEM in the new database.

Results

The new tenant database is created and possibly started, and appears in *Manage Databases*. It is now also in the M_DATABASES view (`SELECT * FROM "PUBLIC"."M_DATABASES"`.

Delivery units (DUs) containing automated content start to be deployed in the background. If the system is online, you can monitor the progress of deployment by executing the following statement:

```
SELECT * FROM "PUBLIC"."M_SERVICE_THREADS" WHERE THREAD_TYPE = 'ImportOrUpdate
Content';
```

For more information about automated content, see *SAP HANA Content* in the *SAP HANA Security Guide*.

Next Steps

- Perform a full data backup. For more information, see *Performing Backups* in the *SAP HANA Administration Guide*.
- Adjust the value for the maximum number of asynchronous I/O requests by updating the value of the `fs.aio-max-nr` parameter in `/etc/sysctl.conf`. For more information, see *Linux Kernel Parameters* in the *SAP HANA Administration Guide*.
- Configure the new tenant database as required. For more information, see the section on managing multitenant database containers in the *SAP HANA Administration Guide*.

Related Information

Delete a Tenant Database [page 80]

Stop and Start a Tenant Database [page 79]
Monitoring Multitenant Resources in SAP HANA Cockpit [page 68]

4.5 Disable Features on a Tenant Database

To safeguard and/or customize your system, certain features of the SAP HANA database can be disabled in tenant databases. You can do this in the SAP HANA cockpit.

Prerequisites

- The system database is registered in the SAP HANA cockpit.
- You have the system privilege INIFILE ADMIN.

Context

Some features of the SAP HANA database are not required or desirable in certain environments, in particular features that provide direct access to the file system, the network, or other resources. To maximize your control over the security of your system, you can disable these features in tenant databases, for example import and export operations or the ability to back up the database.

The system view M_CUSTOMIZABLE_FUNCTIONALITIES provides information about those features that can be disabled and their status. This view exists in both the SYS schema of every database, where it contains database-specific information, and in the SYS_DATABASES schema of the system database, where it contains information about the enablement of features in all databases.

For more information about the features that can be disabled and why, see *Restricted Features in SAP HANA Multitenant Database Containers* in the *SAP HANA Security Guide*.

You disable features in tenant databases in the `customizable_functionalities` section of the `global.ini` file.

> **Note**
>
> All features are enabled in the system database and cannot be disabled.

Procedure

1. Determine which feature(s) you want to disable by referring to the view M_CUSTOMIZABLE_FUNCTIONALITIES (SYS) of the system database.

2. On the *Overview* page of the system database in the SAP HANA cockpit, open *Configuration of System Properties* by clicking the corresponding administration link.
3. Select the configuration file `global.ini` file and the section `customizable_functionalities`.
4. Add a new parameter for the feature that you want to disable:
   a. Specify the database on which you want to blacklist the properties.
   b. In the *Key* field, enter the name of feature that you want to disable and set the value to **`false`**.

   > **Note**
   >
   > If you want to disable the feature on all tenant databases (including any that will be created in the future), enter **`false`** as the system layer value.

5. Repeat for further features not required in the tenant database(s).
6. Restart the affected tenant database(s).

Results

The feature is disabled. You can verify this in the view M_CUSTOMIZABLE_FUNCTIONALITIES (SYS_DATABASES).

Tenant database administrators can see which features are enabled in their database using the view M_CUSTOMIZABLE_FUNCTIONALITIES (SYS).

Related Information

Stop and Start a Tenant Database [page 79]
Restricted Features in Multitenant Database Containers [page 31]

4.6 Enable and Configure Cross-Database Access

In multiple-container systems, read-only queries between tenant databases are supported but not enabled by default. You must first enable this feature for the system in the system database and then configure which databases may communicate with one another. You can do this in the SAP HANA cockpit.

Prerequisites

- The system database is registered in the SAP HANA cockpit.
- You have the system privilege INIFILE ADMIN.

Context

Every tenant database in a multiple-container system is self-contained with its own isolated set of database users and isolated database catalog. However, to support for example cross-application reporting, cross-database SELECT queries are possible. This means that database objects such as tables and views can be local to one database but be read by users from other databases in the same system.

So, for example, the following query would be possible:

```
SELECT *
FROM schema1.table1 AS tab1, db2.schema2.table2 as tab2
WHERE tab2.column2 = 'foobar'
```

For more information about which object types on remote databases can be accessed using this mechanism and which local object types can access remote database objects, see *Cross-Database Access*.

To allow queries between databases, you must first enable cross-database access and then specify which databases may communicate with one other. You can do this by configuring the `global.ini` configuration file in the SAP HANA cockpit.

Procedure

1. On the *Overview* page of the system database in the SAP HANA cockpit, open *Configuration of System Properties* by clicking the corresponding administration link.
2. Select the configuration file `global.ini` file and the section `cross_database_access`.
3. Enable cross-database access by setting the parameter `enabled` to **`true`** on the system layer.
4. Enable communication from one tenant database to one or more other tenant databases as follows:
   a. Add a new parameter `targets_for_<source_db_name>`.
   b. As the value for this parameter, enter the tenant databases that the first database can access as a comma-separated list.

   Example

   You have two databases DB1 and DB2 and you want to be able to access DB1 from DB2. So you add the parameter `targets_for_DB2` with the value **`DB1`**.

   Note

   Cross-database access is configured only in one direction. If in the above example you also want DB2 to be able to access DB1, you would have to add the parameter `targets_for_DB1` with the value **`DB2`**.

Results

Cross-database queries are now possible between the configured databases.

Next Steps

Create remote identities for those users who require cross-database access. For more information, see *Cross-Database Authorization in Multitenant Database Containers* in the *SAP HANA Security Guide*.

In order for a user in one database to be able to run a query or create an object that references objects in another database, the user must be mapped to a sufficiently privileged user in the remote database.

Related Information

Cross-Database Access [page 11]
Troubleshooting Error Situations Related to Cross-Database Access [page 57]
Cross-Database Authorization in Multitenant Database Containers [page 28]
SAP HANA Security Guide

4.6.1 Workload Management and Cross-Database Queries

Cross-database queries are executed on one or more databases. The workload management settings of the tenant database executing the query or part of the query are applied.

To balance and manage different types of workload in SAP HANA (OLAP, OLTP, mixed, and internal), it is possible to classify workloads based on user and application context information and apply resource limitations (for example, a statement memory limit). Workload classes allow SAP HANA to influence dynamic resource consumption at the session or statement level.

The execution of any plan operations of a cross-database query in a remote tenant database is subject to the resource limitations of the workload classes and mappings defined in the remote database. If multiple remote tenant databases are involved in query execution, then different limitations may apply to different portions of the execution plan.

For more information about workload management using workload classes and workload mappings, see *Workload Management* in the *SAP HANA Administration Guide*.

Related Information

SAP HANA Administration Guide

4.6.2 Troubleshooting Error Situations Related to Cross-Database Access

If you are using cross-database access to query data from other tenant databases in your system, some error situations may arise.

Situation 1

You are creating views, procedures, or synonyms to access objects on other tenant databases in the same system. After dropping and re-creating an object on a remote tenant database, you can no longer access the view or procedure on the local tenant database. You get error messages such as `invalidated view` or `invalidated procedure`. You also notice that the IS_VALID column in the system views VIEWS and PROCEDURES do not accurately reflect the fact that the view or procedure is invalid. In addition, there are entries missing in the OBJECT_DEPENDENCIES system view for the affected views, procedures, or synonyms.

What's the problem?

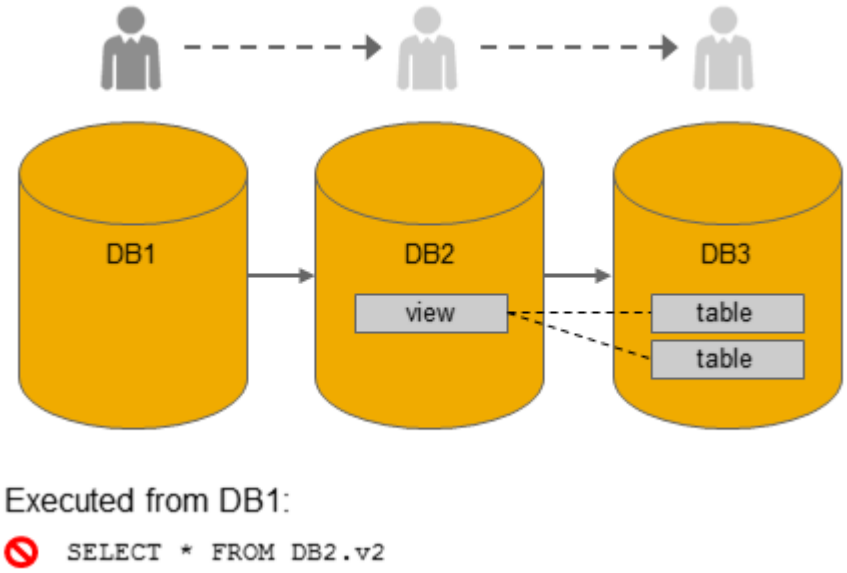
Cross-database access supports only read-only operations. Changes to an object on one tenant database cannot therefore be reflected accurately on other tenant databases that contain objects dependent on the changed object. This affects the validity flag in the relevant system views, as well as the object dependencies. Remote objects may stay valid if they retain their internal object identifier during re-creation and are re-created in an compatible way, but they will become invalid if their internal object identifier changes.

What can I do?

You need to re-create the dependent object in the local tenant database in order for it to become valid again.

Situation 2

You are querying an SQL view or a calculation view on a remote tenant database and the view itself accesses objects on a third tenant database (multi-level cross-database access). You are getting error messages such as `insufficient privilege: not authorized`. Analytic privileges on the third tenant database may be evaluated based on the wrong database user.



What's the problem?

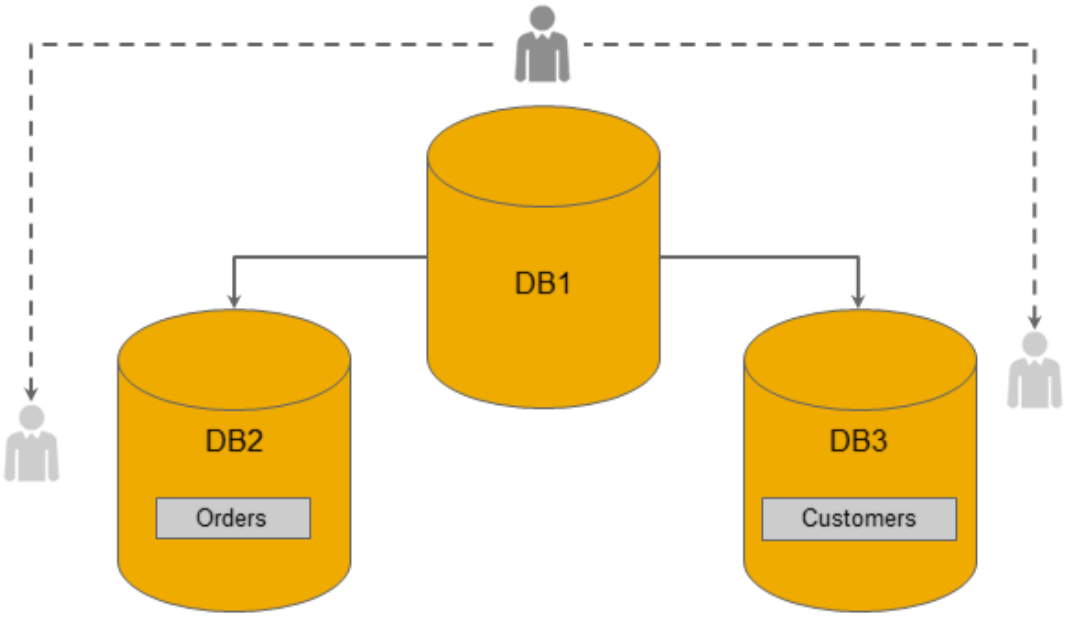
Cross-database queries do not support multiple tenant database levels as part of a view hierarchy, even if communication between databases is enabled (including the required authorized remote users).

What can I do?

Nothing. The feature is not supported.

Situation 3

Your system is running in high isolation mode. Queries that involve more than one remote tenant database run into timeouts. You are getting error messages such as `execution plan aborted` or `current operation canceled by request and transaction rolled back`. Accessing objects on remote tenant databases individually works fine.



Executed from DB1:

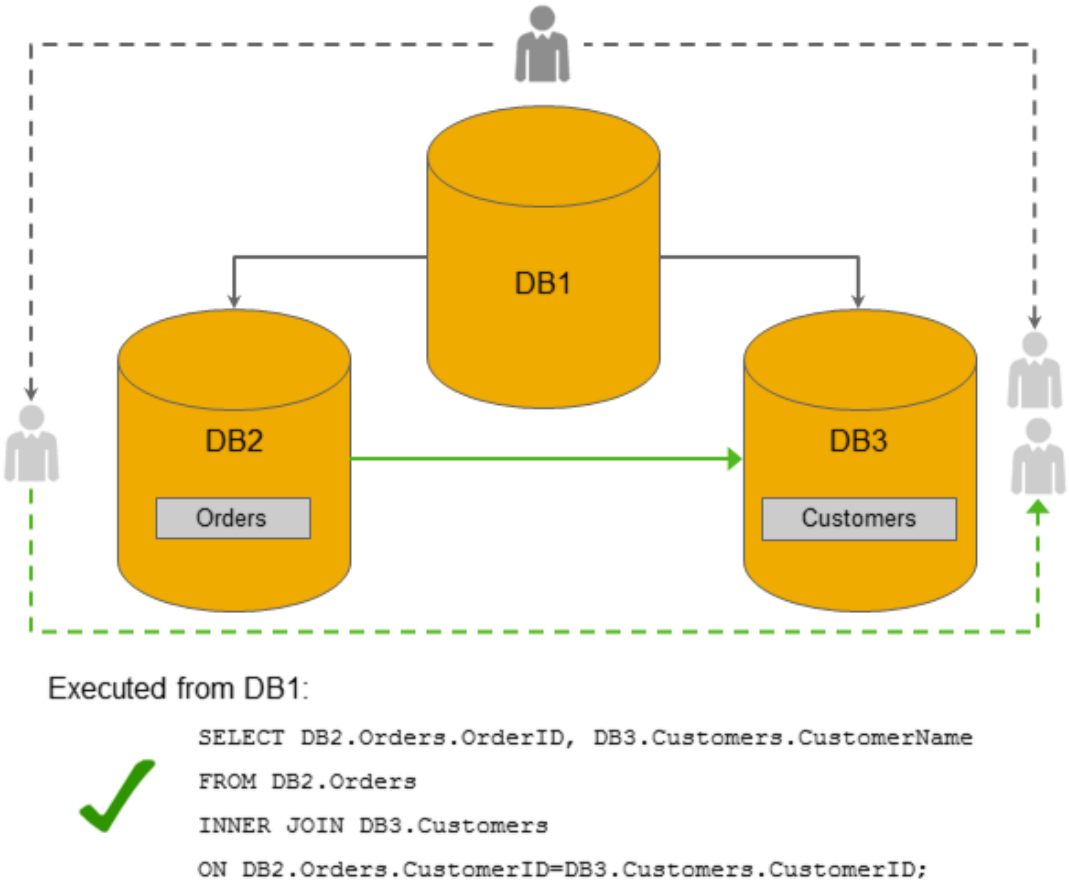
```
SELECT DB2.Orders.OrderID, DB3.Customers.CustomerName
FROM DB2.Orders
INNER JOIN DB3.Customers
ON DB2.Orders.CustomerID=DB3.Customers.CustomerID;
```

What's the problem?

The communication channels that are enabled for cross-database queries are applied in a strict fashion to the underlying network channels as well. This means that one tenant database can only open a network connection to another tenant database if communication between these two databases has been explicitly enabled. The execution plan for a query that involves objects from multiple tenant databases could however lead to direct network connections between any of the tenant databases, even if communication between them has not been explicitly enabled. This specifically applies to joins between tables on two different remote tenant databases.

What can I do?

You need to enable communication between all tenant database pairs that can potentially be involved in a query (including authorized remote users). For more information about how to do this, see *Enable and Configure Cross-Database Access*.



Related Information

Enable and Configure Cross-Database Access [page 54]

4.7 Prevent Changes to System Properties in Tenant Databases

To ensure the stability and performance of the overall system or for security reasons, you can prevent certain system properties from being changed by tenant database administrators, for example, properties related to resource management. A configuration change blacklist is available for this purpose. You configure the blacklist in the SAP HANA cockpit.

Prerequisites

- The system database is registered in the SAP HANA cockpit.
- You have the system privileges INIFILE ADMIN.

Context

System configuration (*.ini) files have a database layer to facilitate the configuration of system properties for individual tenant databases. However, it may be desirable to prevent changes to certain properties being made directly in tenant databases because they could for example affect the performance of the system as a whole (CPU and memory management properties).

For this reason, a dedicated configuration change blacklist, `multidb.ini`, is available. This blacklist contains several critical properties by default. You can customize the default configuration, as well as add further properties by editing the file in the SAP HANA studio.

> **Note**
>
> Properties in the blacklist can still be configured at all levels in the system database. For more information about configuring system properties, see *Configuring SAP HANA System Properties (INI Files)*.

Procedure

1. On the *Overview* page of the system database in the SAP HANA cockpit, open *Configuration of System Properties* by clicking the corresponding administration link.
2. Select the configuration file `multidb.ini` and the section `readonly_parameters`.
3. Add a new parameter to the blacklist:
   a. Specify on which layer you want to blacklist the properties.

      You can choose from the following layers:

      Table 6:

Layer	Result
System	Configuration not possible in any tenant database.
Database	Configuration not possible in the specified tenant database(s)

      > **Note**
      >
      > Layered configuration is possible. A lower-layer configuration overrides a higher-layer configuration. This also allows you to change the default configuration of the blacklist. The example below shows you how you could do this.

   b. In the *Key* field, enter the ini file section that contains the properties you want to blacklist.

If the section exists in more than one configuration file, you can specify the exact configuration file by entering **<file>/<section>**. If you do not specify a configuration file, the properties will be blacklisted in all files that contain the section.

For example, to specify the `communication` section in all configuration files, enter **communication**. But to specify the `communication` section in the xsengine.ini file only, enter **xsengine.ini/communication**.

c. In the *Value* field, enter the properties that you want to blacklist.

   If you want to add all the properties in the section, enter *****. If you want to add all the properties in all sections of a specific file, enter **<filename>/*** (for example, **xsengine.ini/***).

d. Choose *OK*.

e. Add further parameters as required.

Results

Tenant database administrators cannot change the properties in the configuration change blacklist. If they try, they will get the error message: `Change not allowed for tenant database`. System administrators can still change the properties in the system database in all layers.

Example

Layered Configuration

The property `[sql] sql_executors` is blacklisted for all tenant databases in all configuration files by default. You could create a layered configuration for example as follows:

- You change the `sql` entry at the system layer and enter `plan_cache_size` as the value. This overrides the default configuration so that `[sql] plan_cache_size` is blacklisted instead of `[sql] sql_executors`.
- You change the `sql` entry at the system layer and enter `sql_executors` and `plan_cache_size` as the value. This overrides the default configuration so that both `[sql] plan_cache_size` and `[sql] sql_executors` are blacklisted.
- You add a new entry `indexserver.ini/sql` at the system layer with the value `plan_cache_size` as the value. This adds a specific configuration for the `indexserver.ini` file. Here, now only `[sql] plan_cache_size` is blacklisted.

4.8 Configure HTTP(S) Access to Multitenant Database Containers via SAP HANA XS Classic

To enable Web-based applications to send HTTP(S) requests to multitenant database containers via the SAP HANA XS classic server, the internal SAP Web Dispatcher must be configured so it knows which requests to dispatch to which database on the basis of DNS alias host names. You do this by specifying the public URL of every tenant database in the `xsengine.ini` configuration file.

Prerequisites

- You are logged on to the system database.
- You have the system privilege INIFILE ADMIN.
- The network administrator has defined an alias hostname in your organization's Domain Name System (DNS) for every tenant database in the SAP HANA system. The alias hostname must refer to the hostname of the machine that is used for HTTP(S) access to the tenant database.
- You have the role `sap.hana.xs.wdisp.admin::WebDispatcherAdmin`. This is required to access the SAP HANA Web Dispatcher Administration tool for configuring HTTPS.

Context

The XS classic server allows Web-based applications to access SAP HANA via HTTP(S). The internal Web Dispatcher of the SAP HANA system manages these incoming HTTP(S) requests. To allow applications to send requests to specific databases in a multiple-container system, every tenant database needs an alias host name. Requests to the alias host name can then be forwarded to the XS server of the corresponding tenant database. Requests with the physical host name in the HTTP host header are forwarded to the XS server running on the system database.

The default HTTP ports are used in all cases, that is, 80`<instance>` (HTTP) and 43`<instance>` (HTTPS). Alias host names are mapped to internal HTTP(S) ports so that incoming requests can be routed to the correct database.

You configure HTTP(S) access to tenant databases by specifying in the `xsengine.ini` file the URLs by which each tenant database is publicly accessible. The system then automatically configures the Web Dispatcher by generating the required profile entries in the `webdispatcher.ini` configuration file. It is not necessary to specify the URL of the system database, this is done automatically.

> **Note**
>
> This automatic configuration of the Web Dispatcher is controlled by the parameter `[profile] wdisp/system_auto_configuration` in the `webdispatcher.ini` configuration file. If this parameter is set to **`false`** or is not available (revisions earlier than SPS 10), you need to configure the `webdispatcher.ini` file manually.

For HTTPS access, you must subsequently configure the required client certificates and trust stores using the SAP Web Dispatcher Administration tool. The following approaches are supported:

- Using a single "wildcard" server certificate in a single trust store that covers all databases in the system
  Wildcard certificates are more flexible when tenant databases are frequently added and deleted. However, if you use a wildcard certificate, either the server requires its own sub-domain or you must ensure that the certificate cannot be abused from other servers.

  > **Caution**
  >
  > Do not use a wildcard server certificate if strict isolation between tenant databases is required. If authentication relies on a wildcard certificate and a shared trust store, users of one tenant database will be able to log on to other databases in the system.

- Using individual certificates in individual trust stores for each database (as of SPS 11)
  Individual certificates for each database are more suitable in a flat domain structure for individual servers. They also ensure strict isolation between tenant databases. However, they involve more administrative effort to maintain.

Procedure

1. Specify the public URLs of all tenant databases in the xsengine.ini file in one of the following ways:

Option	Description
SAP HANA studio	1. Open the Administration editor and choose the *Configuration* tab. 2. Navigate to the `xsengine.ini` file and expand the `public_urls` section. 3. For each tenant database in the system, add the new properties `http_url` and `https_url` at the **database layer** and enter its public URL as the value: **`http://<virtual_hostname>:80<instance>`**
SQL	For each tenant database, execute the statements: ○ ALTER SYSTEM ALTER CONFIGURATION ('xsengine.ini', 'database', '`<tenant_DB_name>`') SET ('public_urls', 'http_url') = 'http://`<virtual_hostname>`:80`<instance>`' WITH RECONFIGURE; ○ ALTER SYSTEM ALTER CONFIGURATION ('xsengine.ini', 'database', '`<tenant_DB_name>`) SET ('public_urls', 'https_url') = 'https://`<virtual_hostname>`:43`<instance>`' WITH RECONFIGURE;

> **Note**
>
> The following values are set at the **default layer** and represent the URLs of the system database:
>
> ○ **`http://$(SAPLOCALHOST):80$(SAPSYSTEM)`**
>
> ○ **`https://$(SAPLOCALHOST):43$(SAPSYSTEM)`**
>
> By default, the system database initially retrieves any request with the port 80`<instance_no>`. However, as soon as you configure the URLs of tenant databases, it is available under http://`<localhost>`:80`<instance>` only, and not the fully qualified domain name (FQDN). The local host is known to SAP HANA without the FQDN.
>
> If you want to change this default behavior and configure a different URL for the system database, you can do so by executing the following statement:
>
> ```
> ALTER SYSTEM ALTER CONFIGURATION ('nameserver.ini', 'system')
> SET('public_urls', 'http_url') = 'http://<virtual_hostname>:80<instance>' WITH
> RECONFIGURE;
> ```

New entries are now created in the `webdispatcher.ini` file at the host layer for every database. You can verify this by executing the following statement (from the system database):

```
SELECT KEY, VALUE, LAYER_NAME FROM SYS.M_INIFILE_CONTENTS WHERE FILE_NAME =
'webdispatcher.ini' AND SECTION = 'profile' AND KEY LIKE 'wdisp/system%'
```

This returns the following result for example:

```
KEY          | VALUE                                              | LAYER_NAME
wdisp/system_0      | GENERATED, SID=SYS, EXTSRV=http://localhost:30014,
SRCVHOST='myhost'              | DEFAULT
```

```
wdisp/system_1      | GENERATED, SID=MYD, EXTSRV=http://localhost:30042,
SRCVHOST='mydatabase.example.com'     | HOST
```

2. Optional: Secure incoming communication by configuring HTTPS.

Option	Description
Single certificate for all databases	1. Start the SAP HANA Web Dispatcher Administration tool at http://`<localhost>`:80`<instance>`/sap/hana/xs/wdisp/admin/. 2. For the default `SAPSSLS.pse` trust store, create a new SSL key pair and certificate request: 1. From the main menu, choose *SSL and Trust Configuration* > *PSE Management*. 2. From the *Manage PSE* menu, choose *SAPSSLS.pse*. 3. Choose *Recreate PSE*. 4. Enter a distinguished name that matches the host name of all tenant databases. **Example** ○ Physical host name: myhost.example.com ○ Tenant host name 1: mydatabase1.example.com ○ Tenant host name 2: mydatabase2.example.com In this case, you specify **`CN=*.example.com`** as the DN, thus creating a server certificate that matches all tenant databases and the system database. 5. Choose *Create*. 6. Create a certificate request and submit to your certificate authority (CA) for signing (*Create CA Response*). 3. Import the signed certificate For more information, see *Configure HTTPS (SSL) for Client Application Access*.
Individual certificates for each database	1. Start the SAP HANA Web Dispatcher Administration tool at http://`<localhost>`:80`<instance>`/sap/hana/xs/wdisp/admin/. 2. For each tenant database and the system database, create a new trust store with a unique certificate: 1. From the main menu, choose *SSL and Trust Configuration* > *PSE Management*. 2. On the PSE management screen, choose *Create New PSE*. 3. Enter a file name for the new PSE. **Example** `example.pse` 4. Enter the distinguished name: **`CN=<host name used for the tenant database in the public_urls section of the xsengine.ini file>`** 5. Choose *Create*. 6. For the new PSE, create a certificate request and submit to your CA for signing (*Create CA Response*). 7. Import the signed certificate into the new PSE (*Import CA Response*). 3. Configure the Web Dispatcher to use multiple certificates: 1. In the `webdispatcher.ini` file, create or change the parameter `[profile] icm/ssl_config_0`, specifying as the value: **`ID=ssl_config_main, CRED=SAPSSLS.pse, SNI_CREDS=<semicolon (';') separated list of database PSE files>`**

Option	Description
	2. Add **`,SSLCONFIG=ssl_config_main`** to the value of the `icm/server_port` parameter for the HTTPS port (by default `icm/server_port_1`). **Example** `icm/server_port_1 = PROT=HTTPS,PORT=4443$(SAPSYSTEM),PROCTIMEOUT=600, SSLCONFIG=ssl_config_main`

Results

You can access the XS server of tenant databases via the configured URLs.

Tip

If you experience slow response times when accessing the XS server of a tenant database (for example, Web-based applications running on the tenant database), this indicates that the server is not able to resolve the host name correctly using the DNS and retries repeatedly. If this is the case, contact your network administrator for a detailed problem analysis.

As a workaround, you can manually override virtual host name resolution on the machine where the browser is running by modifying the `/etc/hosts` file on the local machine. In this file, append a new line, starting with the static IP address of the server, followed by the virtual host name of your tenant database, for example, "10.20.30.40 mydatabase.example.com". To edit this file you need admin or root privileges.

Next Steps

Optional: Enable access to Web-based applications from the SAP HANA studio.

Some Web-based tools are accessible from the SAP HANA studio, for example, the SAP HANA cockpit and SAP HANA Lifecycle Management tool. If you want to be able to access these tools from a tenant database registered in the studio, you must specify the alias hostname in the properties. You can do this as follows:

1. In the *Systems* view, right-click the tenant database and choose *Properties*.
2. Open the *XS Properties* page and enter the alias hostname in the *XS Host* field.

Related Information

Using SAP Web Dispatcher for Load Balancing with Tenant Databases [page 134]

5 Monitoring and Managing Tenant Databases

To ensure the overall performance and stability of a multiple-container SAP HANA system, you as the system administrator can monitor all tenant databases in the system using the system database. You also can perform administration tasks such as stopping and starting tenant databases, or adding and removing services.

> **Note**
>
> Administration of tenant databases is possible using the SAP HANA cockpit. However, command-line tools are required for some tasks.

Support Model

The following is the general approach for analyzing and resolving issues in tenant databases:

1. Tenant database administrators analyze issues in their tenant databases using the available diagnosis and trace files.
2. If tenant database administrators discover issues that they cannot analyze using diagnosis and trace files, they contact the system administrator.
3. The system administrator can first check the health of the tenant database in the system database by analyzing the monitoring data available in the SYS_DATABASES schema.
4. If the system administrator cannot see what the problem is from the system database, the tenant database administrator needs to provide him with the necessary privileges to access the tenant database directly so that the system administrator can analyze the issue there.

Related Information

Administration of Multitenant Database Containers [page 14]
Stop and Start a Tenant Database [page 79]
Add a Service to a Tenant Database [page 82]
Remove a Service from a Tenant Database [page 84]
Delete a Tenant Database [page 80]
View Diagnosis Files of an Unavailable Tenant Database [page 85]

5.1 Monitoring Multitenant Resources in SAP HANA Cockpit

As the system administrator, you can monitor the availability, resource usage, and performance of tenant databases in the SAP HANA cockpit from the system database.

Aggregate system information is available on the *Overview* page of the system database, while the *Manage Databases* page provides you with a breakdown for all tenant databases. From *Manage Databases*, you can then drill down to more detailed information about each individual tenant database.

> **Note**
>
> The *Overview* page of the system database also displays information and links for the monitoring the system database as a resource itself. For more information, see the section on managing a single resource.

5.1.1 Monitor Status and Resource Usage of a Multitenant Resource

Monitor the overall availability, resource usage, and performance of tenant databases from the system database in the SAP HANA cockpit.

Prerequisites

- To be able to drill down to the detailed information of a tenant database, you must have registered it as a resource in the cockpit.
- To perform operations on a tenant database, you have the system privilege DATABASE ADMIN.

Procedure

1. Open the *Overview* page of the system database in the SAP HANA cockpit.
   In the *System Monitoring and Administration* area, you see the overall system health.

   The alerts tile shows the number of high and medium alerts currently raised in tenant databases.
2. Click the *Overall System Status* tile to open the *Manage Databases* page.
   On the *Manage Databases* page, you can monitor the status and resource usage of individual databases, as well as perform other administration tasks such as stopping and starting tenant databases and creating new tenant databases.

   If you want to examine a particular database in more detail, you can drill down further on the *Manage Databases* page by clicking the aspect you're interested in (for example, database name, alerts or used memory). For more information about drill-down options, see *Database Details*.

Depending on the situation, you may need to perform further operations on a tenant database (stop, start, and delete).

Related Information

Database Details [page 69]

5.1.1.1 Database Details

The *Manage Databases* page provides you with detailed information about all databases in a multiple-container system, as well as several drill-down options for more detailed information about individual databases.

The table below lists the information available for databases, as well as the available drill-down option.

Table 7:

Column	Description	Drill-Down Option
Database Name	Name of the tenant database	Click the database name to open the *Manage Services* page *Manage Services* allows you to analyze the status and resource usage of the individual services of the database. For more information, see *Service Details*. From *Manage Services*, you can also stop and start services.
Status	Status of the tenant database: • *Running* • *Not running* • *Starting* • *Stopping*	No drill-down available

Column	Description	Drill-Down Option
Alerts	The number of high and medium priority alerts in the database	Click the number of alerts to open them on the *Alerts* page *Alerts* allows you to view and analyze alerts occurring in the database. You can view past occurrences of alerts. For more information, see *Alert Details* and *Alert Priorities*. You can also see how alerts are configured. For more information, see *Alert Checker Details* and *Alert Checker Statuses*. **Note** Only those alerts that identify situations with a potentially system-wide impact are visible, for example, the physical memory on a host is running out. Alerts that expose data in the tenant database (for example, table names) are **not** visible to the system administrator in the system database.
Start Time	The time of the most recent start of the database	No drill-down available
Used Memory	The used memory of the database in relation to the system	Click the used memory bar to open the *Performance Monitor* app The *Performance Monitor* app allows you to visually analyze historical performance in the database across a range of related performance indicators. For more information, see *Key Performance Indicators*.
CPU Usage	The CPU usage of the database in relation to the system	Click the CPU usage bar to open the *Performance Monitor* app The *Performance Monitor* app allows you to visually analyze historical performance in the database across a range of related performance indicators. For more information, see *Key Performance Indicators*.
Disk Usage	The disk usage of the database in relation to the system	Click the disk usage bar to open the *Performance Monitor* app The *Performance Monitor* app allows you to visually analyze historical performance in the database across a range of related performance indicators. For more information, see *Key Performance Indicators*.

Related Information

Service Details [page 71]
Alert Details [page 76]
Alert Priorities [page 77]
Alert Checker Details [page 77]
Alert Checker Statuses [page 78]
Key Performance Indicators [page 73]

5.1.1.2 Service Details

Manage Services provides you with detailed information about database services for an individual resource.

> **Note**
>
> Not all of the columns listed below are visible by default. You can add and remove columns in the table personalization dialog, which you open by clicking the personalization icon in the table toolbar.

The table below lists the information available for services.

Table 8:

Column	Description
Host	Name of the host on which the service is running
Status	The status of the service The following statuses are possible: • *Running* • *Running with Issues* • *Not Running* To investigate why the service is not running, you can navigate to the crash dump file created when the service stopped. **Note** The crash dump file opens in the *Trace* tool of the SAP HANA Web-based Development Workbench. For this, you need the role `sap.hana.xs.ide.roles::TraceViewer` or the parent role `sap.hana.xs.ide.roles::Developer`.
Service	Service name, for example, indexserver, nameserver, xsengine, and so on

Column	Description
Role	Role of the service in a failover situation Automatic failover takes place when the service or the host on which the service is running fails. The following values are possible: • *Master* The service is the active master worker. • No entry The service is a slave worker. • *Standby* The service is in standby mode. It does not contain any data and does not receive any requests.
Port	Port that the system uses for internal communication between services
Start Time	Time at which the service started **Note** The time is given in the timezone of the SAP HANA server.
CPU	Mini chart visualizing the CPU usage of the service Clicking the mini chart opens the *Performance Monitor* for a more detailed breakdown of CPU usage.
Memory	Mini chart visualizing the memory usage of the service Clicking the mini chart opens the *Memory Allocation Statistics* app for a more detailed breakdown of memory usage.
Used Memory (MB)	Amount of memory currently used by the service Clicking the mini chart opens the *Memory Allocation Statistics* app for a more detailed breakdown of memory usage.
Peak Memory (MB)	Highest amount of memory ever used by the service
Effective Allocation Limit (MB)	Effective maximum memory pool size that is available to the process considering the current memory pool sizes of other processes
Memory Physical on Host (MB)	Total memory available on the host
All Process Memory on Host (MB)	Total used physical memory and swap memory on the host
Allocated Heap Memory (MB)	Heap part of the allocated memory pool
Allocated Shared Memory (MB)	Shared memory part of the allocated memory pool
Allocation Limit (MB)	Maximum size of allocated memory pool
CPU Process (%)	CPU usage of process
CPU Host (%)	CPU usage on host

Column	Description
Memory Virtual on Host (MB)	Virtual memory size on the host
Process Physical Memory (MB)	Process physical memory used
Process Virtual Memory (MB)	Process virtual memory
Shrinkable Size of Caches (MB)	Memory that can actually be freed in the event of a memory shortage
Size of Caches (MB)	Part of the allocated memory pool that can potentially be freed in the event of a memory shortage
Size of Shared Libraries (MB)	Code size, including shared libraries
Size of Thread Stacks (MB)	Size of service thread call stacks
Used Heap Memory (MB)	Process heap memory used
Used Shared Memory (MB)	Process shared memory used
SQL Port	SQL port number
Process ID	Process ID

5.1.1.3 Key Performance Indicators

The *Performance Monitor* allows you select a range of host-level and service-level KPIs to analyze historical performance data of the SAP HANA database.

Host KPIs

Table 9:

KPI	Description
CPU	CPU used by all processes related to the operating system (OS)
Database resident memory	Physical memory used by all SAP HANA database processes
Total resident memory	Physical memory used by all OS processes
Physical memory size	Total physical memory
Database used memory	Memory used by all SAP HANA database processes
Database allocation limit	Memory allocation limit for all SAP HANA database processes
Disk used	Disk space used by data, log, and trace files belonging to the SAP HANA database
Disk size	Total disk size
Network in	Bytes read from the network by all processes
Network out	Bytes written to the network by all processes

KPI	Description
Swap in	Bytes read from swap memory by all processes
Swap out	Bytes written to swap memory by all processes

Services KPIs

Table 10:

KPI	Description
CPU	CPU used by the database process
System CPU	CPU used by the database process relative to the operating system
Memory used	Memory used by the database process
Memory allocation limit	Effective allocation limit of the database process
Handles	Number of open handles in the index server process
Ping time	Indexserver ping time including nsWatchdog request and collection of service-specific KPIs
Swap in	Bytes read from swap by the process
Open connections	Number of open SQL connections
Open transactions	Number of open SQL transactions
Blocked transactions	Number of blocked SQL transactions
Statements	Number of finished SQL statements
Active commit ID range	Range between newest and oldest active commit ID
Active transaction ID range	Range between newest and oldest active transaction ID
Pending session request count	Number of pending requests
Active versions	Number of active MVCC versions
Acquired record locks	Number of acquire record locks
Read requests	Number of read requests (selects)
Write requests	Number of write requests (insert, update, and delete)
Merge requests	Number of merge requests
Column unloads	Number of table and column unloads
Active threads	Number of active threads
Waiting threads	Number of waiting threads
Total threads	Total number of threads
Active SqlExecutors	Total number of active SqlExecutor threads
Waiting SqlExecutors	Total number of waiting SqlExecutor threads
Total SqlExecutors	Total number of SqlExecutor threads

KPI	Description
Data write size	Bytes written to data area
Data write time	Time used for writing to data area
Log write size	Bytes written to log area
Log write time	Time used for writing to log area
Data read size	Bytes read from data area
Data read time	Time used for reading from data area
Log read size	Bytes read from log area
Log read time	Time used for reading from log area
Data backup write size	Bytes written to data backup
Data backup write time	Time used for writing to data backup
Log backup write size	Bytes written to log backup
Log backup write time	Time used for writing to log backup

5.1.2 Monitor Alerts for a Multitenant Resource

Alert situations in tenant databases may potentially impact the health of the overall system. For this reason, you as system administrator can monitor alerts occurring in individual tenant databases. You can do this from the system database in the SAP HANA cockpit.

Prerequisites

To be able to drill down to the alert information of a tenant database, you must have registered it as a resource in the cockpit.

Procedure

1. On the *My Resources* page or any group overview page, click the name of a system database (on the *Top Resources with Alerts* tile or the *Recently Accessed* tile, for example) to open that system database's*Overview* page.
   In the *System Monitoring and Administration* area, you see the overall system health, as well as aggregate alert information.
2. Open the alerts of a particular database by clicking the number of alerts indicated for that database.

   All high and medium priority alerts occurring in all databases in the system are displayed in list format on the left. To see more detailed information about a specific alert on the right, simply select it.

> **Tip**
>
> You can also access database-specific alerts from *Alerts* in the *Manage Databases* app.

Next Steps

It may be helpful to see how alerts are configured in individual tenant databases. To navigate to the configuration of alert checkers from the *Alerts* app, click *View Alert Configuration* in the footer toolbar.

5.1.2.1 Alert Details

When you select an alert, detailed information about the alert is displayed on the right.

The following detailed information about an alert is available:

Table 11:

Detail	Description
Category	The category of the alert checker that issued the alert Alert checkers are grouped into categories, for example, those related to memory usage those related to transaction management and so on.
Next Scheduled Run	When the related alert checker is next scheduled to run If the alert checker has been switched off (alert checker status *Switched Off*) or it failed the last time it ran (alert checker status *Failed*), this field is empty because the alert checker is no longer scheduled.
Interval	The frequency with which the related alert checker runs If the alert checker has been switched off (alert checker status *Switched Off*) or it failed the last time it ran (alert checker status *Failed*), this field is empty because the alert checker is no longer scheduled.
Alerting Host & Port	Name and port of the host that issued the alert In a system replication scenario, alerts issued by secondary system hosts can be identified here. This allows you to ensure availability of secondary systems by addressing issues before an actual failover. For more information about monitoring secondary systems in SAP HANA, see *Monitoring Secondary Sites* in the *SAP HANA Administration Guide*.
Alert Checker	Name and description of the related alert checker
Proposed Solution	Possible ways of resolving the problem identified in the alert, with a link to the supporting app, if available
Past Occurrences of Alert	Configurable graphical display indicating how often the alert occurred in the past

Related Information

SAP HANA Administration Guide

5.1.2.2 Alert Priorities

The priority of an alert indicates the severity of the problem and how quickly action needs to be taken.

Table 12:

Priority	Description
Information	Action recommended to improve system performance or stability
Low	Medium-term action required to mitigate the risk of downtime
Medium	Short-term action required (few hours, days) to mitigate the risk of downtime
High	Immediate action required to mitigate the risk of downtime, data loss, or data corruption

5.1.2.3 Alert Checker Details

When you select an alert checker *Alert Configuration*, detailed information about the alert checker and its configuration is displayed on the right.

The following detailed information about an alert checker is available:

Table 13:

Detail	Description
Header information	The name of the alert checker, its status, and the last time it ran
Description	Description of what the alert checker does, for example what performance indicator it measures or what setting it verifies
Alert Checker ID	The unique ID of the alert checker
Category	The category of the alert checker Alert checkers are grouped into categories, for example those related to memory usage, those related to transaction management, and so on.

Detail	Description
Threshold Values for Prioritized Alerting	The values that trigger high, medium, low, and information alerts issued by the alert checker The threshold values and the unit depend on what the alert checker does. For example, alert checker 2 measures what percentage of disk space is currently used so its thresholds are percentage values. **Note** Thresholds can be configured for any alert checker that measures variable values that should stay within certain ranges, for example, the percentage of physical memory used, or the age of the most recent data backup. Many alert checkers verify only whether a certain situation exists or not. Threshold values **cannot** be configured for these alert checkers. For example, alert checker 4 detects services restarts. If a service was restarted, an alert is issued.
Interval	The frequency with which the alert checker runs
Schedule Active	Indicator of whether the alert checker is running automatically at the configured interval
Proposed Solution	Possible ways of resolving the problem identified by the alert checker

Related Information

Alert Checker Statuses [page 78]

5.1.2.4 Alert Checker Statuses

The status of an alert checker indicates whether it is running on schedule, it has failed and been disabled by the system, or you switched it off.

Table 14:

Status	Description
Active	The alert checker is running on schedule.

Status	Description
Failed	The alert checker failed the last time it ran (for example due to a shortage of system resources), so the system disabled it. The alert checker remains disabled for a specific length of time before it is automatically re-enabled. This length of time is calculated based on the values in the following columns of the table STATISTICS_SCHEDULE (_SYS_STATISTICS): • INTERVALLENGTH • SKIP_INTERVAL_ON_DISABLE Once INTERVALLENGTH x SKIP_INTERVAL_ON_DISABLE has elapsed, the alert checker is re-enabled. The default values for all alert checkers are such that failed checkers remain disabled for 1 hour. The system determines the status of every alert checker and/or whether the time to re-enablement has elapsed every 60 seconds. You can also re-enable the alert checker manually by switching it back on in *Alert Configuration*.
Switched Off	You switched off the alert checker schedule. If you want the alert checker to run again automatically, you must manually switch it back on.

5.1.3 Stop and Start a Tenant Database

You stop and start tenant databases from the system database on the *Manage Databases* page of the SAP HANA cockpit.

Prerequisites

- You have the system privilege `DATABASE ADMIN`.
- If you are stopping the database, consider backing it up first.

Context

As a system administrator, you can start tenant databases either individually, or all at once by starting the whole system. The same applies for stopping tenant databases. For more information about how to stop and start the whole system, see the sections on stopping and starting a resource.

Procedure

1. Open *Manage Databases* by drilling down from the *Overall System Status* tile on the *Overview* page of the system database in the SAP HANA cockpit.
2. Select the tenant database that you want to stop or start.
3. Choose *Stop Tenant Database* or *Start Tenant Database* in the footer toolbar.
   The database starts stopping or starting. This may take a few moments.

Results

- The database is started or stopped.

  > **Note**
  >
  > If you stopped the database, it is a hard stop. The database is stopped immediately even if users are connected. Open transactions are aborted and rolled back; no savepoint operation is forced. It is not possible to back up a stopped database.

- The status of database changes accordingly.

Related Information

SAP HANA SQL and System Views Reference
SAP HANA Administration Guide

5.1.4 Delete a Tenant Database

You can delete tenant databases that are no longer required. You delete tenant databases from the system database using the *Manage Databases* app of the SAP HANA cockpit.

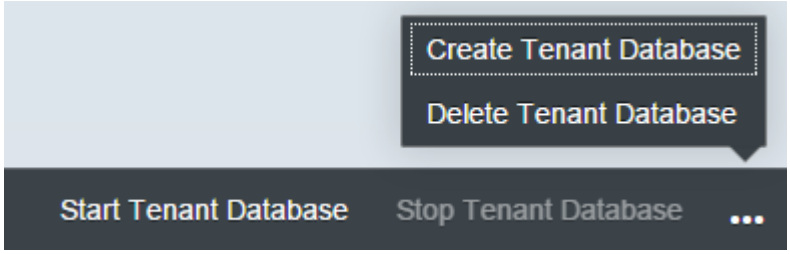
Prerequisites

You have the system privilege `DATABASE ADMIN`.

Procedure

1. Open *Manage Databases* by drilling down from the *Overall System Status* tile on the *Overview* page of the system database in the SAP HANA cockpit.
2. Stop the tenant database that you plan to delete by selecting it and then clicking *Stop Tenant Database* in the footer toolbar.

   The system starts stopping the database. Once stopped, its status changes to *Not running*.
3. In the footer toolbar, open the overflow menu and choose *Delete Tenant Database*.



Results

The system starts deleting the database. Once deleted, it disappears from the list of databases. Volumes and trace files are removed.

Next Steps

If you configured the SAP Web Dispatcher to route HTTP(s) requests to the deleted database, you need to update the configuration.

Related Information

Configure HTTP(S) Access to Multitenant Database Containers via SAP HANA XS Classic [page 62]
SAP HANA SQL and System Views Reference
Stop and Start a Tenant Database [page 79]

5.2 Add a Service to a Tenant Database

To scale out a tenant database and/or distribute it across multiple hosts, you can add further server components, for example, an additional index server or a separate XS server. You add a service to a tenant database using the ALTER DATABASE statement.

Prerequisites

- You are logged on to the system database.
- You have the system privilege DATABASE ADMIN.

Context

You can add any service that persists data to an existing tenant database. You do this from the system database using the ALTER DATABASE ADD `<service>` statement. The main services that you may need to add are the following:

- indexserver
- xsengine

  > **Note**
  >
  > After database creation, the xsengine service automatically runs embedded in the (master) index server. If you add a separate xsengine service, the embedded service is stopped and removed. If you have already configured the SAP Web Dispatcher for HTTP(s) access based on the embedded xsengine service, you must update the configuration.

- scriptserver
- dpserver

> **Note**
>
> You cannot add a statisticsserver service. This always runs embedded in the master index server of a tenant database.

In addition to the service you want to add, you may also specify the following information:

- The host on which the service is to be added
- The number of the internal communication port of the new service

  > **Note**
  >
  > The default port number range for tenant databases is 3`<instance>`40—3`<instance>`99. This means that the maximum number of tenant databases that can be created per instance is 20. However, you can increase this by reserving the port numbers of further instances. You do this by configuring the property `[multidb] reserved_instance_numbers` in the `global.ini` file. The default value of this

property is 0. If you change the value to 1, the port numbers of one further instance are available (for example, 30040—30199 if the first instance is 00). If you change it to 2, the port numbers of two further instances are available (for example, 30040—30299 if the first instance is 00). And so on.

Caution

Adding a service breaks the backup history of the database. To ensure that the database is fully recoverable, perform a full backup (data backup or storage snapshot) immediately after adding a service.

Procedure

Execute the ALTER DATABASE statement.

Example

Table 15:

What	How
A new index server with automatic host placement and automatic port assignment	`ALTER DATABASE DB0 ADD 'indexserver'`
A new index server on a specific host accessible via a specific SQL port (30148) **Note** The SQL port number is the internal communication port number plus 1.	`ALTER DATABASE DB0 ADD 'indexserver' AT LOCATION 'HOST_A:30146'`
A new index server on a specific host and a new XS server on an automatically selected host	`ALTER DATABASE DB0 ADD 'indexserver' AT 'HOST_B' ADD 'xsengine'`

Results

- New data and log volumes are created on the host and the information is entered in the system landscape information of system database.
- The service is added to the M_SERVICES system view.
- The service is started.

Next Steps

- If you added an xsengine service, update the configuration of the SAP Web Dispatcher. For more information, see *Configure HTTP Access to Multitenant Database Containers*.

- Perform a full backup (data backup or storage snapshot).

Related Information

Configure HTTP(S) Access to Multitenant Database Containers via SAP HANA XS Classic [page 62]
http://help.sap.com/hana/SAP_HANA_SQL_and_System_Views_Reference_en.pdf

5.3 Remove a Service from a Tenant Database

You can remove server components that are no longer needed from a tenant database. Removing services may be necessary, for example, if you want to remove a tenant database from a host. You remove a service from a tenant database using the ALTER DATABASE statement.

Prerequisites

- You are logged on to the system database.
- You have the system privilege DATABASE ADMIN.

Context

You can remove any service that persists data from an existing tenant database, with the exception of the master indexserver service. You do this from the system database using the ALTER DATABASE REMOVE `<service>` statement.

You must specify the host and internal communication port number of the service to be removed.

> ⚠ Caution
>
> Removing a service breaks the backup history of the database. To ensure that the database is fully recoverable, perform a full backup (data backup) immediately after removing a service.

If you want to move a tenant database with only a single indexserver service to another host, you can either back up the tenant database and restore it on the new host, or move the tenant.

Procedure

Execute the ALTER DATABASE statement.

Example

```
ALTER DATABASE DB0 REMOVE 'indexserver' AT LOCATION 'HOST_A:30146'
```

Note

The port number specified is that of the internal communication port of the service being removed.

Results

- The service is stopped and removed from the system landscape information of system database.
- The service is removed from the M_SERVICES system view.
- Data volumes and traces files are removed.

Next Steps

- If you removed an xsengine service, update the configuration of the SAP Web Dispatcher. For more information, see *Configure HTTP Access to Multitenant Database Containers*.
- Perform a full backup (data backup or storage snapshot).

Related Information

Configure HTTP(S) Access to Multitenant Database Containers via SAP HANA XS Classic [page 62]
http://help.sap.com/hana/SAP_HANA_SQL_and_System_Views_Reference_en.pdf
Move a Tenant Database to Another System [page 117]

5.4 View Diagnosis Files of an Unavailable Tenant Database

If a tenant database is unavailable, for example because it's stopped or experiencing major performance problems, the tenant database administrator can't access diagnosis files. In this case, you as the system

administrator can access the diagnosis files of the tenant database from the system database using the SAP HANA studio.

Prerequisites

- You are logged on to the system database.
- You have the system privilege CATALOG READ.

Procedure

1. In the SAP HANA studio, open the Administration editor and choose *Diagnosis Files*.
   The diagnosis files of the system database are displayed.
2. Using the *Database* filter, select the tenant database(s) whose diagnosis files you want to see.
   The diagnosis files of the selected tenant database(s) are displayed.

Next Steps

If more detailed diagnosis information is required (for example for SAP Support), you can trigger the collection of a full system information dump for tenant databases. For more information, see *Collecting Diagnosis Information for SAP Support* in the *SAP HANA Administration Guide*.

5.5 Resetting the SYSTEM User Password in Multitenant Database Containers

The system database and all tenant databases in a multiple-container system each have their own SYSTEM user. You can reset the password of these SYSTEM users as the operating system administrator by starting the name server (system database) or index server (tenant database) in emergency mode.

5.5.1 Reset the SYSTEM User Password of the System Database

If the password of the SYSTEM user in the system database is lost, you can reset it as the operating system administrator by starting the name server in emergency mode.

Prerequisites

- You cannot log on to the database as the SYSTEM because the password has been irretrievably lost.

  > **Note**
  >
  > If you can log on as SYSTEM and you want to change the password, do not use the procedure described here. Use the SAP HANA studio or execute the ALTER USER SQL statement directly: `ALTER USER SYSTEM PASSWORD <new_password>`.

- You have the credentials of the operating system administrator (`<sid>`adm).

Procedure

1. Log on to the server on which the name server of the system database is running as the operating system user (that is, `<sid>`adm user).
2. Open a command line interface.
3. Shut down the instance by executing the following command:

   ```
   /usr/sap/<SID>/HDB<instance>/exe/sapcontrol -nr <instance> -function StopSystem HDB
   ```

4. In a new session, start the name server of the system database by executing the following commands:
   - `/usr/sap/<SID>/HDB<instance>/hdbenv.sh`
   - `/usr/sap/<SID>/HDB<instance>/exe/hdbnameserver -resetUserSystem`

   After some start-up notifications, the prompt `resetting of user SYSTEM - new password` appears, followed by additional notifications:

```
:/usr/sap/ /  > /usr/sap/ C/  /exe/hdbnameserver -resetUserSyst
em
Starting interactive mode for resetting user SYSTEM...
unclean shutdown of service instance with pid 29905.
service startup...
accepting requests at
searching for master nameserver  ...
assign as master nameserver. assign to volume 1 started
service startup...
Checking for recovery request ...
Loading topology ...
Opening persistence ...
run as transaction master
Loading topology ...
Loading licensing ...
setStarting(nameserver@ )
setActive(nameserver@ )
service assigned as master
service start as systemserver
setInactive(preprocessor@ )
setInactive(webdispatcher@ )
setInactive(compileserver@ )
setInactive(indexserver@ )
resetting of user SYSTEM - new password:

HDB              HDBSettings.sh   hdbenv.csh       work/
HDBAdmin.sh      backup/          hdbenv.sh        xterms
HDBSettings.csh  exe/             /

HDB              HDBSettings.sh   hdbenv.csh       work/
HDBAdmin.sh      backup/          hdbenv.sh        xterms
HDBSettings.csh  exe/             /

HDB              HDBSettings.sh   hdbenv.csh       work/
HDBAdmin.sh      backup/          hdbenv.sh        xterms
HDBSettings.csh  exe/             /

HDB              HDBSettings.sh   hdbenv.csh       work/
HDBAdmin.sh      backup/          hdbenv.sh        xterms
HDBSettings.csh  exe/             /
NewPassword1
new pw accepted.
(Re)Activating user SYSTEM...
done
```

Figure 7: Reset SYSTEM User Password (System Database)

5. After the appearance of the last notification, enter a new password for the SYSTEM user.

   You must enter a password that complies with the password policy configured for the system.

   The password for the SYSTEM user of the system database is reset and the name server stops.

6. In a new session, start the instance by executing the following command:

   ```
   /usr/sap/<SID>/HDB<instance>/exe/sapcontrol -nr <instance> -function StartSystem
   HDB
   ```

Results

The password of the SYSTEM user of the system database is reset. You have to change the new password the next time you log on with this user.

If you previously deactivated the SYSTEM user, it is now also reactivated. This means you will need to deactivate it again.

5.5.2 Reset the SYSTEM User Password of a Tenant Database

If the password of the SYSTEM user in a tenant database is lost, you can reset it as the operating system administrator by starting the index server in emergency mode.

Prerequisites

- You cannot log on to the database as the SYSTEM because the password has been irretrievably lost.

  > **Note**
  >
  > If you can log on as SYSTEM and you want to change the password, do not use the procedure described here. Execute the ALTER USER SQL statement directly: `ALTER USER SYSTEM PASSWORD <new_password>`.

- You have the system privilege DATABASE ADMIN.
- You have the credentials of the operating system administrator (`<sid>`adm).

Procedure

1. Stop the tenant database by executing the following statement:

   `ALTER SYSTEM STOP DATABASE <database_name>`
2. Log on to the server on which the index server of the tenant database is running as the operating system user (that is, `<sid>`adm user).
3. Open a command line interface.
4. Export the database name and then start the index server by executing the following commands:
   - `/usr/sap/<SID>/HDB<instance> export DBNAME=<database_name>`
   - `/usr/sap/<SID>/HDB<instance>/hdbenv.sh`
   - `DBNAME=<database_name> /usr/sap/<SID>/HDB<instance>/exe/hdbindexserver -port <internal port> -resetUserSystem`

   The following prompt appears: `resetting of user SYSTEM - new password`

5. Enter a new password for the SYSTEM user.
   The password for the SYSTEM user of the tenant database is reset and the index server stops.
6. Restart the tenant database:

```
ALTER SYSTEM START DATABASE <database_name>
```

Results

The password of the SYSTEM user of the tenant database.

You have to change the new password the next time you log on with this user. If you previously deactivated the SYSTEM user, it is now also reactivated. This means you will need to deactivate it again.

6 Configuring Memory and CPU Usage in Multiple-Container Systems

Manage and control the memory and CPU usage of your multiple-container system by configuring limits for individual tenant databases. If necessary, you can also reserve memory for the system database.

Managing Resource Usage of Tenant Databases

Several system properties allow you to influence the allocation of memory and CPU resources in SAP HANA systems. System properties (INI) files have a database layer to facilitate the configuration of properties for individual tenant databases.

The properties listed below are particularly useful for influencing the resource consumption of tenant databases. For more information about additional options, see the sections referenced under *Related Information*.

- `[memorymanager] allocationlimit` in the `global.ini` file
  Use this property to limit the maximum amount of memory (in MB) that can be allocated individually to processes of a tenant database. Each process of a tenant database can allocate the specified value. Setting the allocation limit too low might cause the tenant database to become inaccessible until more memory can be allocated.

  Example

  Executed from the system database:`ALTER SYSTEM ALTER CONFIGURATION ('global.ini', 'DATABASE', 'MYDB') SET ('memorymanager', 'allocationlimit') = '8192' WITH RECONFIGURE;`

  Note

  Memory alignment will happen on the fly and may therefore take some time. To make it happen immediately, you can restart the database.

- `[execution] max_concurrency` in the `global.ini` file
  Use this property to influence the maximum number of CPU cores that can be used for each tenant database by limiting the number of concurrently running threads used by the JobExecutor subsystem. A reasonable default value is the number of cores divided by the number of tenant databases. Do not specify a value of `0`. A change of this value takes effect immediately.

  Example

  Executed from the system database:`ALTER SYSTEM ALTER CONFIGURATION ('global.ini', 'DATABASE', 'MYDB') SET ('execution', 'max_concurrency') = '4' WITH RECONFIGURE;`

> **Note**
>
> In NUMA architectures, setting the `max_concurrency` parameter is not enough to achieve the desired performance gains, so you should also bind sockets that share memory using the affinity setting. For more information, see *Controlling CPU Consumption*.

Managing Memory Usage of System Database

After installation, the system database contains only data required to monitor and manage the system, as well as statistics data related to itself. This results in an average memory consumption of 15 GB.

However, if the system database is experiencing performance problems, for example, out-of-memeory situations, you can reserve a minimum amount of memory (MB) for the system database by configuring the parameter `[multidb] systemdb_reserved_memory` in the `global.ini` file.

Related Information

Parameter Reference: Memory Consumption [page 92]
Controlling Parallel Execution of SQL Statements [page 93]
Controlling CPU Consumption [page 95]

6.1 Parameter Reference: Memory Consumption

The `memorymanager` section of the `global.ini` file contains parameters that allow you to control the memory consumption of SAP HANA.

You can change the default settings in the configuration editor of the SAP HANA studio (recommended) or directly in the **global.ini** system properties file.

These parameters require tracking to be enabled in global.ini [resource_tracking]. Resource tracking creates a runtime overhead in SAP HANA:

- enable_tracking = on
- memory_tracking = on

> **Note**
>
> In a system that supports multitenant database containers, you can configure the `global.ini` at both the system level and the database level. Parameters configured at the system level apply to the complete system and all databases. Parameters configured at the database level apply to the specified database only.

`global_allocation_limit` - limits the amount of memory that can be used by the system as a whole.

- The parameter `global_allocation_limit` defines the maximum memory allocation limit in GB.
- The global allocation limit is calculated by default as follows: 90% of the first 64 GB of available physical memory on the host plus 97% of each further GB. Or, in the case of small physical memory, physical memory minus 1 GB. A missing entry or a value of 0 results in the system using the default settings.
- Does not require a restart. Available since SPS 08.

> **Note**
>
> In a system that supports multitenant database containers, the global allocation limit configured at the system layer of the `global.ini` file is always effective regardless of any value configured at the database layer.

`statement_memory_limit` - defines the maximum memory allocation per statement in GB.

- When the statement memory limit is reached, a dump file is created with "compositelimit_oom" in the name. The statement is aborted, but otherwise the system is not affected.
- The default value is 0 (no limit). Set this parameter to a value between 1 GB and the value of the global allocation limit.
- Does not require a restart (applies to new statements). Available since SPS 09.

`statement_memory_limit_threshold` - defines the maximum memory allocation per statement as a percentage of the global allocation limit.

- If a value for this parameter has been set then the statement memory limit is only applied if the current SAP HANA memory consumption exceeds the statement memory limit threshold as a percentage of the global allocation limit.
- The default value is 0% (the statement_memory_limit is always respected). Set this parameter to a value between 1 GB and the value of the global allocation limit.
- Does not require a restart (applies to new statements). Available since SPS 09.

6.2 Controlling Parallel Execution of SQL Statements

You can apply ini file settings to control the two thread pools SQLExecutors and JobExecutors that control the parallelism of statement execution.

> **Caution**
>
> The settings described here should only be modified when other tuning techniques like remodeling, repartitioning, and query tuning have been applied. Modifying the parallelism settings requires a thorough understanding of the actual workload since they have impact on the overall system behavior. Modify the settings iteratively by testing each adjustment. For more information, see *Understand your Workload*.

On systems with highly concurrent workload, too much parallelism of single statements may lead to sub-optimal performance. Note also that partitioning tables influences the degree of parallelism for statement execution; in general, adding partitions tends to increase parallelism. You can use the parameters described in this section to adjust the CPU utilization in the system.

Two thread pools control the parallelism of the statement execution:

- SqlExecutor
  This thread pool handles incoming client requests and executes simple statements. For each statement execution, an SqlExecutor thread from a thread pool processes the statement. For simple OLTP-like statements against column store as well as for most statements against row store, this will be the only type of thread involved. With OLTP we mean short running statements that consume relatively little resources, however, even OLTP-systems like SAP Business Suite may generate complex statements.
- JobExecutor
  The JobExecutor is a job dispatching subsystem. Almost all remaining parallel tasks are dispatched to the JobExecutor and its associated JobWorker threads.

For both SqlExecutor and JobExecutor, a separate limit can be set for the maximum number of threads. For example, on a system where OLAP workload would normally consume too many CPU resources you can apply a maximum value to reserve resources for OLTP workload.

> ⚠ Caution
>
> Lowering the value of these parameters can have a drastic effect on the parallel processing of the servers and reduce the performance of the overall system. Adapt with caution by iteratively making modifications and testing. For more information, see *Understand your Workload* and the following subsection, *Parameter Reference: CPU*.

Parameters for SQL Executor

The following SqlExecutor parameters are in the `sql` section of the `indexserver.ini`.

- `sql_executors` - The target number of threads that can be used.
- `max_sql_executors` - The maximum number of threads that can be used. Not set by default so that new threads are created to handle incoming requests.

Parameters for JobExecutor

The following JobExecutor parameters are in the `execution` section of the `global.ini` or `indexserver.ini`.

- `max_concurrency` - The target number of threads that can be used. Note that from SPS12, this parameter should only need to be modified for multitenant database container installations.
- `max_concurrency_hint` - Limit concurrency hint even if more active job workers would be available.
- `default_statement_concurrency_limit` - Used to restrict the actual degree of parallel execution per connection within a statement.

JobExecutor settings do not solely affect OLAP workload, but also other SAP HANA subsystems (for example, memory garbage collection, savepoint writes). The JobExecutor executes also operations like table updates and backups, which were delegated by the SqlExecutor. JobExecutor settings are soft limits, meaning the JobExecutor can “loan” threads (this applies to the SQL Executor as well), if available, and then fall back to the maximum number of threads when done.

➡ Tip

In a system that supports multitenant database containers, a reasonable default value for the `max_concurrency` parameter is the number of cores divided by the number of tenant databases. Do not specify a value of **0**.

6.3 Controlling CPU Consumption

If the physical hardware on a host is shared between several processes you can use CPU affinity settings to assign a set of logical cores to a specific SAP HANA process. These settings are coarse-grained and apply on the OS and process-level.

Prerequisites

Using this workload management option we firstly analyze how the system CPUs are configured and then, based on the information returned, apply affinity settings in `daemon.ini` to bind specific processes to logical CPU cores. Processes must be restarted before the changes become effective. This approach applies primarily to the use cases of SAP HANA multitenant database containers and multiple SAP HANA instances on one server; you can use this, for example, to partition the CPU resources of the system by tenant database.

➡ Tip

As an alternative to applying CPU affinity settings you can achieve similar performance gains by changing the parameter `[execution] max_concurrency` in the `indexserver.ini` configuration file. This may be more convenient and does not require the system to be offline. For more information, see *Managing Resources in Multiple-Container Systems*.

To make the changes described here you require access to the operating system of the SAP HANA instance to run the Linux `lscpu` command and you require the privilege INIFILE ADMIN.

Context

For Xen and VmWare, the users in the VM guest system see what is configured in the VM host. So the quality of the reported information depends on the configuration of the VM guest. Therefore SAP cannot give any performance guarantees in this case.

Procedure

1. Firstly, to confirm the physical and logical details of your CPU architecture, analyze the system using the `lscpu` command. This command returns a listing of details of the system architecture. The table which follows gives a commentary on the most useful values based on an example system with 2 physical chips (sockets) each containing 8 physical cores. These are hyperthreaded to give a total of 32 logical cores.

#	Feature	Example Value
1	Architecture	x86_64
2	CPU op-mode(s)	32-bit, 64-bit
3	Byte Order	LittleEndian
4	CPUs	32
5	On-line CPU(s) list	0-31
6	Thread(s) per core	2
7	Core(s) per socket	8
8	Socket(s)	2
9	NUMA node(s)	2
21	NUMA node0 CPU(s)	0-7,16-23
22	NUMA node1 CPU(s)	8-15,24-31

   - 4-5: This example server has 32 logical cores numbered 0 - 31
   - 6-8: Logical cores ("threads") are assigned to physical cores. Where multiple threads are assigned to a single physical core this is referred to as 'hyperthreading'. In this example, there are 2 sockets, each socket contains 8 physical cores (total 16). Two logical cores are assigned to each physical core, thus, each core exposes two execution contexts for the independent and concurrent execution of two threads.
   - 9: In this example there are 2 NUMA (Non-uniform memory access) nodes, one for each socket. Other systems may have multiple NUMA nodes per socket.
   - 21-22: The 32 logical cores are numbered and specifically assigned to one of the two NUMA nodes.

   Note

   Even on a system with 32 logical cores and two sockets the assignment of logical cores to physical CPUs and sockets can be different. It is important to collect the assignment in advance before making changes. A more detailed analysis is possible using the system commands described in the next step. These provide detailed information for each core including how CPU cores are grouped as siblings.

2. In addition to the `lscpu` command you can use the set of system commands in the `/sys/devices/system/cpu/` directory tree. For each logical core there is a numbered subdirectory beneath this node (`/`

`cpu12/` in the following examples). The examples show how to retrieve this information and the table gives details of some of the most useful commands available:

Example

```
cat /sys/devices/system/cpu/present

cat /sys/devices/system/cpu/cpu12/topology/thread_siblings_list
```

Command	Example Output	Commentary
present	0-15	The number of logical cores available for scheduling.
cpu12/topology/core_siblings_list	4-7, 12-15	The cores on the same socket.
cpu12/topology/thread_siblings_list	4,12	The logical cores assigned to the same physical core (hyperthreading).
cpu12/topology/physical_package_id	1	The socket of the current core - in this case cpu12.

3. Based on the results returned you can restrict CPU usage of SAP HANA processes to certain CPUs or ranges of CPUs. You can do this for the following servers: nameserver, indexserver, compileserver, preprocessor and xsengine (each server has a section in the daemon.ini file). The examples and commentary below show the syntax for the ALTER SYSTEM CONFIGURATION commands required. The changed affinity settings only takes effect after a restart of the affected SAP HANA processes.

   Other Linux commands which are relevant here are `sched_setaffinity` and `numactl`: `sched_setaffinity` limits the set of CPU cores available (by applying a CPU affinity mask) for execution of a specific process (this could be used, for example, to isolate tenants in a MDC) and `numactl` controls NUMA policy for processes or shared memory.

Example

To restrict the nameserver to two logical cores of the first CPU of socket 0 (see line 21 in the example above), use the following affinity setting:

```
ALTER SYSTEM ALTER CONFIGURATION ('daemon.ini', 'SYSTEM') SET
('nameserver', 'affinity') = '0,16'
```

Example

To restrict the preprocessor and the compileserver to all remaining cores (that is, all except 0 and 16) on socket 0 (see line 21 in the example above), use the following affinity setting:

```
ALTER SYSTEM ALTER CONFIGURATION ('daemon.ini', 'SYSTEM') SET
('preprocessor', 'affinity') = '1-7,17-23'
ALTER SYSTEM ALTER CONFIGURATION ('daemon.ini', 'SYSTEM') SET
('compileserver', 'affinity') = '1-7,17-23'
```

Example

To restrict the indexserver to all cores on socket 1 (see line 22 in the example above), use the following affinity setting:

```
ALTER SYSTEM ALTER CONFIGURATION ('daemon.ini', 'SYSTEM') SET
('indexserver', 'affinity') = '8-15,24-31'
```

Example

To set the affinity for two tenant databases called DB1 and DB2 respectively in a multitenant database container setup, use the following affinity settings::

```
ALTER SYSTEM ALTER CONFIGURATION ('daemon.ini', 'SYSTEM') SET
('indexserver.DB1', 'affinity') = '1-7,17-23';
ALTER SYSTEM ALTER CONFIGURATION ('daemon.ini', 'SYSTEM') SET
('indexserver.DB2', 'affinity') = '9-15,25-31';
```

Related Information

Configuring Memory and CPU Usage in Multiple-Container Systems [page 91]

7 Copying and Moving Tenant Databases Between Systems

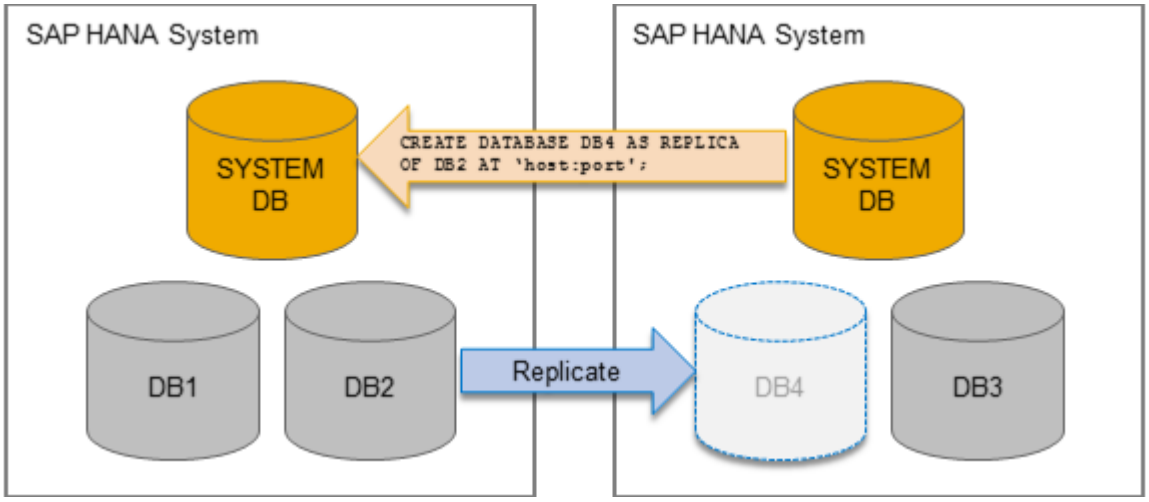
Using SAP HANA system replication mechanisms, SAP HANA multitenant database containers can be copied and moved securely and conveniently from one SAP HANA system to another with near-zero downtime. This allows you to respond flexibly to changing resource requirements and to manage your system landscape efficiently.

The following sections provide an overview of copying or moving a tenant database using system replication.

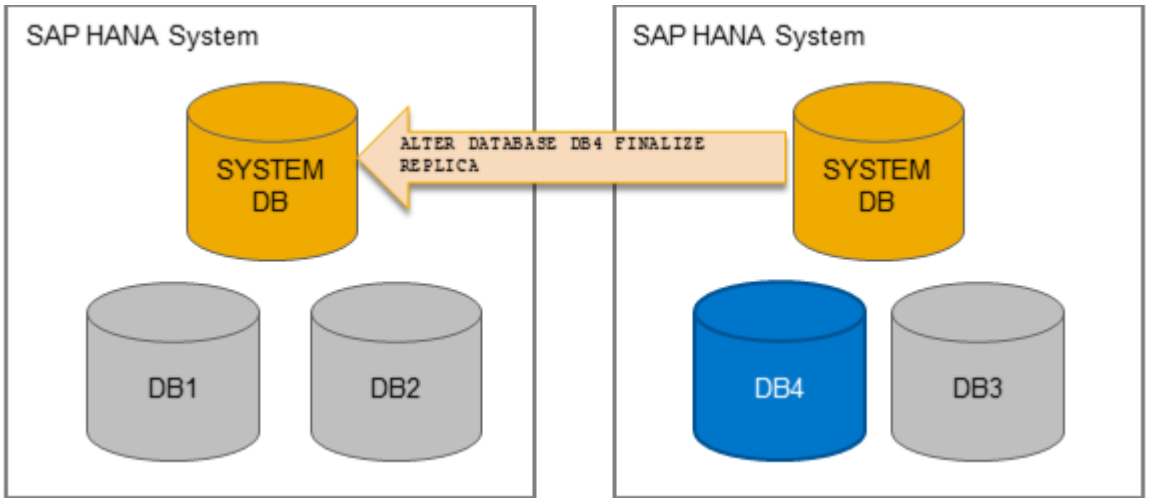
- Process Overview [page 99]
- Use Cases [page 100]
- Which Data Is Copied or Moved? [page 101]
- Recoverability After Copy or Move [page 102]
- General System Prerequisites [page 102]
- Other Copy and Move Methods [page 103]

Process Overview

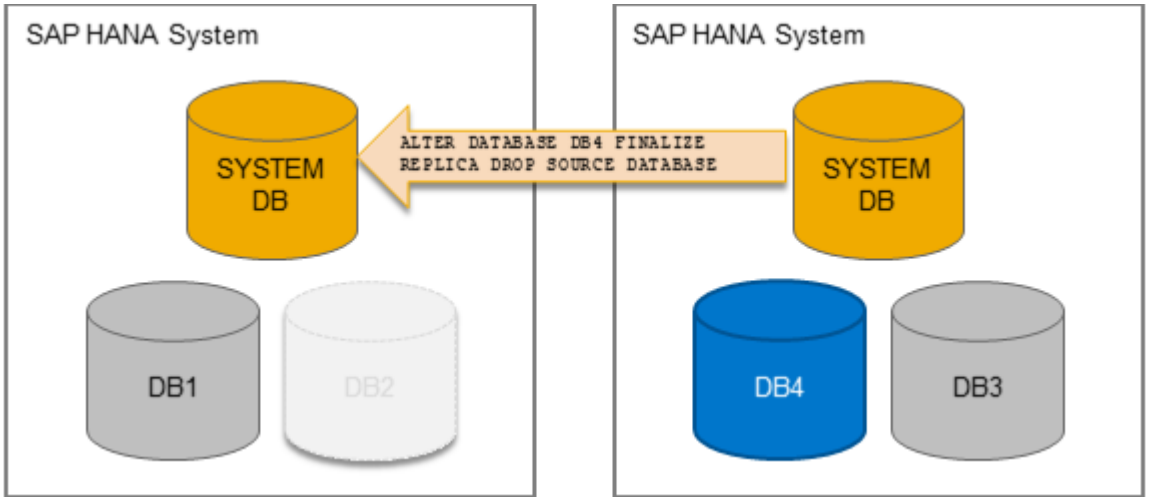
Copying and moving a tenant database are essentially the same process. First, the tenant database is copied through the replication of all of its data to a newly created tenant database in a target system:



Once all data has been successfully transferred, the new tenant database is started as a separate, independent database:



If the aim is to move the tenant database to the new system, the original tenant database is deleted and the new tenant database takes over:



The only difference between copying and moving a tenant database therefore is what happens to the original tenant database after all data has been transferred to the new tenant database in the target system.

In both cases, the new tenant database starts running as a fully separate, independent database.

Several tenant databases can be copied or moved to a system at the same time. It is also possible to copy or move a tenant database to a system with a different isolation level to the source system.

Use Cases

Copying and moving a tenant database from one system to another in this way has several applications, including:

- Load balancing between systems

For example, a tenant database is running a more demanding workload than anticipated, so you move it to a system running on a host with more CPU resources.

- Management of deployment environment
  For example, you want to copy a tenant database running in your test system to the live production system.
- Tenant-database-specific upgrades
  For example, you want to upgrade a single tenant database but not the entire system, so you move the tenant database to a system already running the higher version.
- Template databases
  For example, you create a tenant database with a default configuration that you want to reuse as the basis for new tenant databases in other systems. You can simply copy the tenant database as a template to other systems.

Which Data Is Copied or Moved?

When a tenant database is copied or moved, data is replicated from the original tenant database to the new tenant database in the target system.

The following table indicates which types of data are replicated and which are not.

Table 16:

Type of Data	Replicated?
Data and logs of the tenant database	Yes
Trace and log files	No
Data backups	No
Configuration (*.ini) files with tenant-database-specific values This refers to files in the directory `$DIR_INSTANCE/../SYS/global/hdb/custom/config/<database_name>`	No
Certificates and certificate collections stored in the tenant database This refers to the digital certificates and certificate stores used for certificate-based user authentication and secure communication between SAP HANA and JDBC/ODBC clients. **Note** If these certificates are stored in the file system in personal security environments (PSEs), they will not be replicated. To ensure that they are replicated, migrate the file-system-based PSEs to in-database certificate collections before copying or moving the tenant database. For more information about how to do this, see SAP Note 2175664.	Yes
Database-specific root key used for the internal data encryption service in the secure store file system (SSFS) **Note** The root key used for data volume encryption is not replicated.	Yes

Type of Data	Replicated?
Application function libraries	No

Recoverability After Copy or Move

When you copy a tenant database, the new tenant database does not have a backup history and cannot be recovered immediately after being copied. For this reason, it is important to perform a full data backup after you copy.

When you move a tenant database, the backup history of the original tenant database is retained in the new tenant database. As long as data and log backups of the source system are at a location accessible to the target system, the new tenant database is recoverable immediately after the move.

> ⚠ Caution
>
> If you subsequently create a tenant database in the source system with the same name as the moved tenant database, the backup files of the original database are overwritten.

Prerequisites and Implementation Considerations

- The copy and move process involves the creation of a new tenant database in the target system. Therefore, the target tenant database must not already exist in the target system.
- The target system must have a software version equal to or higher than the source system.
- If data volume encryption is enabled in the original system, data will be decrypted before replication and then re-encrypted (with a new root key) in the new database. However, during the copy and move process, data must be replicated via a secure (SSL/TLS) network connection by default.
- Since the copy and move process uses system replication mechanisms, system replication must not be enabled on either the source or target system for high availability purposes for the entire duration of the copy or move process.
- There can be no changes to the topology of the original tenant database while the move or copy is in progress. In other words, until the copy or move has been finalized, it is not possible to add services to or remove services from the source tenant database.
- If the source system is configured for host auto-failover, the copy or process will fail in the event of failover to a standby host. If this happens, the new tenant database must be deleted on the target system and the copy or move process started again.
- The following components must not be configured in the source tenant database:
  - Rserve server
  - SAP HANA dynamic tiering (extended storage server)
  - SAP HANA accelerator for SAP ASE (extended transaction service)
  - SAP HANA smart data streaming (streaming host)

Other Copy and Move Methods

Backup and Recovery

It is possible to use backup and recovery to copy or move tenant databases between two systems. However, we recommend using SAP HANA system replication as described here. The main advantage of using system replication over backup and recovery is the absence of downtime. Using backup and recovery, you would have to shut down the original database after backing it up until the new database is successfully recovered. This is particularly critical if you are moving a tenant database. System replication is also a more convenient method because you don't need to move files between the different systems.

To copy or move a tenant database within the same system, we recommend using backup and recovery.

SAP HANA Database Lifecycle Manager (HDBLCM)

To copy or clone an entire system, use the SAP HANA database lifecycle manager (HDBLCM) as described in the *SAP HANA Platform Lifecycle Management* section of the *SAP HANA Administration Guide*.

Related Information

SAP Note 2175664
Security of the Copy and Move Process [page 107]
Copy and Move Process [page 103]
SAP HANA Administration Guide

7.1 Copy and Move Process

Understand the stages and steps involved in copying and moving a tenant database from a source system to a target system using SAP HANA system replication.

Overview

The process of copying or moving a tenant database is driven entirely by the target system.

The following figure shows the stages involved, as well as who performs the individual steps in each stage: the system administrator or the target system. Each step is then described in more detail.

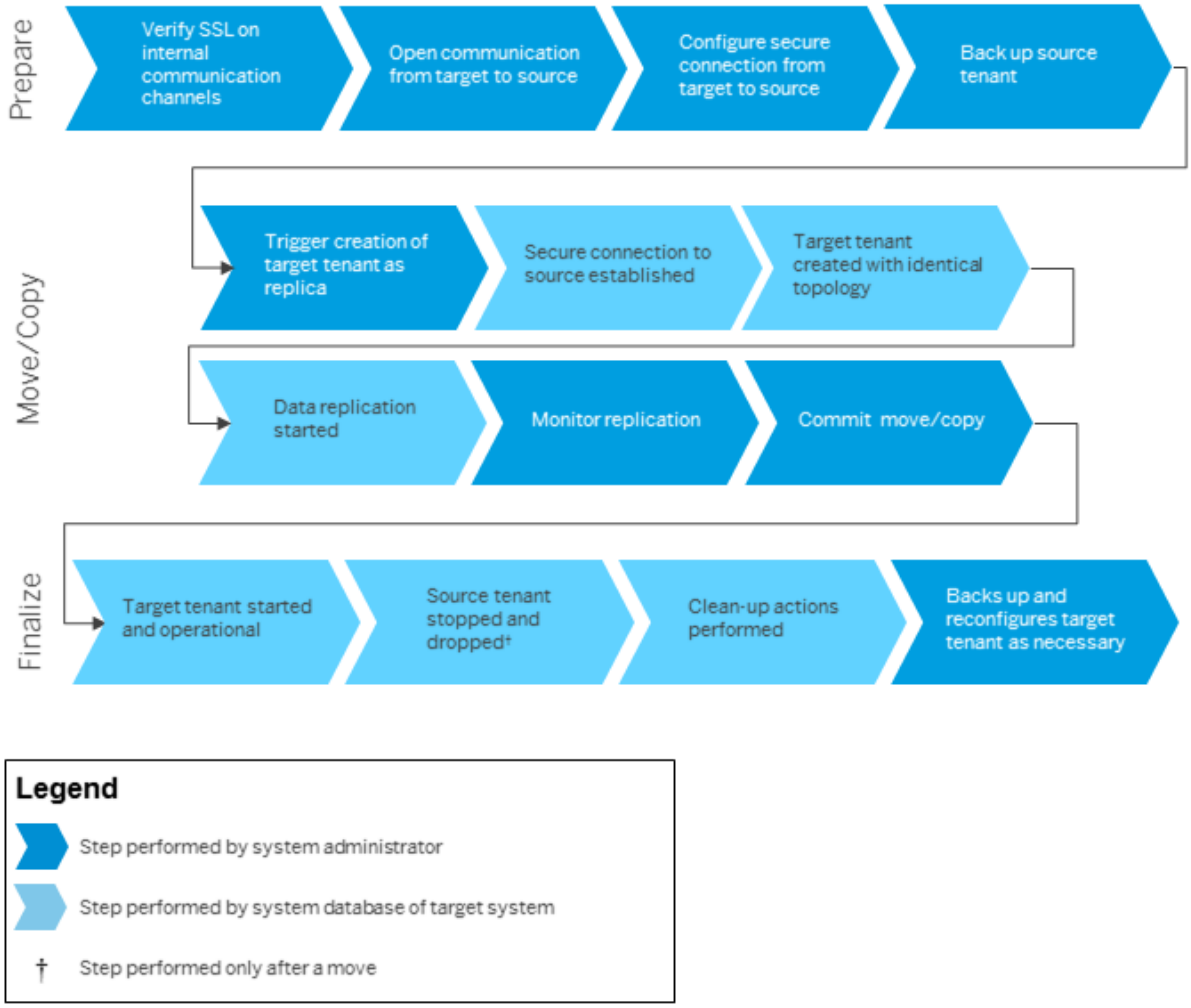


Figure 8: Tenant Move/Copy Process Flow

Prepare

Table 17:

Who?	Does What?	Where?
System administrator	Verifies that TLS/SSL is enabled on internal communication channels	System database of source and target system
	Opens communication from the target system to the source system by enabling source system services to listen on all network interfaces	System database of source system

Who?	Does What?	Where?
	Configures secure connection from target system to source system by: 1. Creating a certificate collection with the purpose `DATABASE REPLICATION` and adding the root certificate of the source system to the new collection This will allow trust to be established between the system databases of the target and source system for external communication via SQL. 2. Creating a credential in the target system to enable authenticated access from the target system to the source system	System database of target system
	Backs up the source tenant database	System database of source system

For more information, see *Preparing to Copy or Move a Tenant Database*.

Copy and Move

Table 18:

Who?	Does What?	Where?
System administrator	Triggers the creation of the tenant database as a replica of the source tenant database by executing the SQL statement `CREATE DATABASE AS REPLICA`	System database of target system
System database of target system	Establishes a secure connection to the system database using the stored credentials created above For subsequent secure communication between the systems, a set of public and private key pairs and public-key certificates is generated in the source system database. These will be used to secure communication from the target system database to the source system database, and from the target tenant database to the source tenant database. The generated key pairs and certificates are imported into two newly created certificate collections in the target system database.	From target system to source system
	Creates a new tenant database with the same topology as the tenant database in the source system	Target system
	Initiates replication of data between the services in the source tenant database and the corresponding services in the target database	From source tenant database to target tenant database
System administrator	Monitors the progress of data replication in system view `SYS_DATABASES.M_DATABASE_REPLICAS`	System database of target system or source system

Who?	Does What?	Where?
	Once the replication status is `ACTIVE` (indicating that all data has been transferred), commits the copy by executing the SQL statement `ALTER DATABASE FINALIZE REPLICA` In the case of a move, the administrator indicates that the source tenant database is to be dropped: `DROP SOURCE DATABASE`.	System database of the target system

For more information, see *Copy a Tenant Database to Another System* and *Move a Tenant Database to Another System*.

Finalize

Table 19:

Who?	Does What?	Where?
System database in target system	Starts the target tenant database	Target system
	If the source tenant database is being moved to the target system, stops and drops the source tenant database	Source system
	Performs clean-up operations: • Deletes any cross-database dependencies to the original tenant database in other tenant databases of the source system (**move only**) • Deletes any remote identity dependencies of users in the new tenant database in the target tenant database (**copy and move**) • Generates a new root key used for data volume encryption and re-encrypts data if data volume encryption is enabled (**copy and move**)	Source system and tenant database in target system
System administrator	Performs manual post-copy or post-move steps: • Back up the target tenant database This is only necessary after a copy since the new tenant database does not have a backup history and cannot be recovered. After a move, the new tenant database has the backup history of the original tenant database and can be recovered if data and log backups of the source system are at a location accessible to the target system. • Reverse preparatory steps required to secure the copy process • If necessary, reconfigure parameters in *.ini files with tenant-database-specific values • If necessary, reconfigure cross-database access	System database of the target system

Related Information

Security of the Copy and Move Process [page 107]
Preparing to Copy or Move a Tenant Database [page 109]
Copy a Tenant Database to Another System [page 115]
Move a Tenant Database to Another System [page 117]

7.2 Security of the Copy and Move Process

Copying or moving a tenant database from one system to another is a secure end-to-end process.

Secure Network Communication

The copy and move process ensures end-to-end data encryption and host authentication on the basis of X.509 client certificates. Dedicated certificates and trust stores (referred to as certificate collections) are created as part of the copy or move process for the purpose of that specific copy or move. Certificates and certificate collections are stored directly in the system databases as database objects.

> **Note**
>
> If secure network communication is not required, it can be disabled. As a result, a copy or move of a tenant database can be initiated without the exchange of certificates. For more information, see *Disable Secure Network Communication*.

For more information about in-database certificate management, see the *SAP HANA Security Guide*.

Authorization and Authentication

The copy and move process is triggered from the system database of the target system by a system administrator. To be able to execute the copy or move statements, the administrator user requires the system privilege `DATABASE ADMIN`.

To be able to establish a connection to the system database of the source system, the target system must be authenticated on the source system. This is achieved through the creation of a credential in the secure internal credential store of the system database of the target system. The required credential must be created manually by an administrator in the system database of the target system before the copy or move is started. For more information about the secure internal credential store, see the *SAP HANA Security Guide*.

Encryption Key Handling

SAP HANA features two data encryption services: data encryption in the persistence layer and an internal data encryption service available to applications requiring data encryption. The instance secure store in the file system (SSFS) is used to protect the root keys for these encryption services.

The root key used for data volume encryption is **changed automatically** in the new tenant database as part of the copy or move operation.

The root key used for the data encryption service is **not changed automatically** in the new tenant database as part of the copy or move operation. This root key is extracted from the SSFS of the original tenant database and replicated to the new tenant database and stored in the instance SSFS of the target system. It is not possible to change this root key manually.

> ⚠ Caution
>
> Do not change the root key manually in the new tenant database. This will result in information in the SSFS and the database becoming inconsistent and encrypted data becoming inaccessible.

Cross-Database Dependencies

If cross-database access is enabled in the original tenant database, some configured dependencies are automatically deleted to ensure no unauthorized communication paths or user mappings can be exploited in the copied or moved tenant database.

Permitted Communication Paths

Part of the configuration of cross-database access involves specifying which tenant databases may communicate with each other and in which direction.

After a tenant database is moved to another system, it is deleted in the source system. However, communication paths referencing it will still exist in one or more of the other tenant databases in the source system. When the move operation is finalized, all such references to the original database are **automatically** deleted in other tenant databases of the source system.

Communication paths configured in the tenant database in the target system must manually be reconfigured after a move or copy.

User Mappings

Another aspect of cross-database access configuration is the mapping of users in one tenant database to users in another tenant database using remote identities.

After a tenant database is moved or copied to another system, some of its database users may still be associated as remote identities for users in other databases in the source system. When the move or copy operation is finalized, all remote identity information of users in the new tenant database is **automatically** deleted.

New user mappings must be manually configured in the tenant database in the target system.

Related Information

Disable Secure Network Communication [page 109]

7.2.1 Disable Secure Network Communication

During the copy and move process, data must be replicated via a secure (SSL/TLS) network connection by default. If this is not necessary in your scenario, you can disable this requirement.

Prerequisites

- You have the system privilege INIFILE ADMIN.
- Neither the source system or target system is configured for high isolation (the value of the `database_isolation` in the `[multidb]` section of the `global.ini` file is **`low`**) .
- The value of parameter `ssl` in the `[communication]` section of the `global.ini` file is set to **`false`**.
- External communication is not configured for SSL in either the source system or target system.

Procedure

1. In the SAP HANA studio, open the *Configuration* tab of the Administration editor.
2. Navigate to the `[multidb]` section of the `global.ini` file.
3. Change the value of the parameter `enforce_ssl_database_replication` to **`off`**.

7.3 Preparing to Copy or Move a Tenant Database

Before you copy or move a tenant database to another system, you must perform several steps. These are primarily to enable the systems to communicate with each other securely.

1. Verify TLS/SSL Configuration of Internal Communication Channels [page 110]
   In both the source system and the target system, verify that TLS/SSL is enabled on internal communication channels on the basis of the system public key infrastructure (system PKI).
2. Open Communication From Target to Source System [page 111]
   Open communication from the target system to the source system by enabling services in the source system to listen on all network interfaces.
3. Set Up Trust Relationship Between Target and Source Systems [page 112]

Create a certificate collection in the system database of the target system and add either the public-key certificate of the system database of source system, or the root certificate of the source system. This certificate is used to secure communication between the systems via external SQL connections.

4. Create Credential for Authenticated Access to Source System [page 113]
   Create a credential to enable authenticated access to the source system for the purpose of copying or moving a tenant database.
5. Back Up Tenant Database [page 114]
   Back up the tenant database that will be copied or moved.

7.3.1 Verify TLS/SSL Configuration of Internal Communication Channels

In both the source system and the target system, verify that TLS/SSL is enabled on internal communication channels on the basis of the system public key infrastructure (system PKI).

Prerequisites

You have a user in the system database of both systems with the system privilege INIFILE ADMIN.

Procedure

> **Remember**
>
> This step must be performed in both the source system and the target system.

1. In the SAP HANA studio, open the *Configuration* tab of the Administration editor.
2. Navigate to the `[communication]` section of the `global.ini` file.
3. Verify that value of the parameter `ssl` is set to **`systemPKI`** at the `SYSTEM` layer.
   If it's not, change the value of the parameter accordingly.
4. Restart the system.

Task overview: Preparing to Copy or Move a Tenant Database [page 109]

Next task: Open Communication From Target to Source System [page 111]

7.3.2 Open Communication From Target to Source System

Open communication from the target system to the source system by enabling services in the source system to listen on all network interfaces.

Prerequisites

You have the credentials of operating system administrator `<sid>`adm for the source system.

Context

Use the SAP HANA database lifecycle manager (HDBLCM) to configure inter-service communication so that the services of the target system can listen on all available network interfaces.

> **Note**
>
> It is only necessary to perform this step in the source system. However, if you later want to be able to monitor the progress of the copy or move operation from the source system, you can also do it in the target system.

Procedure

> **Note**
>
> The following procedure describes how to do this using the Web user interface. For more information about using the command-line interface or graphical user interface of the SAP HANA database lifecycle manager, see the *SAP HANA Administration Guide*.

1. Open the SAP HANA database lifecycle manager by entering the following URL in a browser:
   `https://<host>:1129/lmsl/HDBLCM/<sid>/index.html`
2. Click the tile *Configure Inter-Service Communication*.
3. When prompted, enter the password of the `<sid>`adm user.
4. Select the setting *global*.
5. Click *Run* to apply the new setting.
6. Close the application and log out.

Task overview: Preparing to Copy or Move a Tenant Database [page 109]

Previous task: Verify TLS/SSL Configuration of Internal Communication Channels [page 110]

Next task: Set Up Trust Relationship Between Target and Source Systems [page 112]

7.3.3 Set Up Trust Relationship Between Target and Source Systems

Create a certificate collection in the system database of the target system and add either the public-key certificate of the system database of source system, or the root certificate of the source system. This certificate is used to secure communication between the systems via external SQL connections.

Prerequisites

- You have a user in the system database of the target system with the system privileges CERTIFICATE ADMIN, TRUST ADMIN, and DATABASE ADMIN.
- You have the public-key certificate of the system database of the source system (or the root certificate of the source system) used for external communication.
  If this certificate does not already exist, you can create it using the SAP Web Dispatcher administration tool or the SAPGENPSE tool, both of which are delivered with SAP HANA. If you are using OpenSSL, you can use the tools provided with OpenSSL.
  If the certificate does exist, its location depends on how you manage certificates in your system. Certificates stored in database (recommended) are contained in the certificate store. The required certificate is assigned to the collection with purpose *SSL*. Certificates stored in the file system are contained in tenant database-specific personal security environments or PSEs (default `$SECUDIR/sapsrv.pse`).
  For more information, see *SSL Configuration on the SAP HANA Server* in the SAP HANA Security Guide.

Procedure

1. In the system database of the target system, create a certificate collection with the purpose `DATABASE REPLICATION`.

   You can do this using the *Certificate Collections* app of the SAP HANA cockpit or by executing the following SQL statement:

   ```
   CREATE PSE <collection_name>;
   SET PSE <collection_name> PURPOSE DATABASE REPLICATION;
   ```

2. If necessary, import the public-key certificate of the system database of the source system (or the root certificate of the source system) into the certificate store.

   If the certificate is not already in the certificate store, you can import it using the *Certificate Store* app of the SAP HANA cockpit or by executing the following SQL statement:

   ```
   CREATE CERTIFICATE FROM '<certificate_content>';
   ```

3. Add the system database certificate (or root certificate) to the new collection.

   You can do this using the *Certificate Collections* app of the SAP HANA cockpit or by executing the following SQL statement:

   ```
   ALTER PSE <collection_name> ADD CERTIFICATE <certificate_id>;
   ```

   > **Tip**
   >
   > You will find the certificate ID in the system view `SYS.CERTIFICATES`.

Task overview: Preparing to Copy or Move a Tenant Database [page 109]

Previous task: Open Communication From Target to Source System [page 111]

Next task: Create Credential for Authenticated Access to Source System [page 113]

Related Information

SAP HANA Security Guide

7.3.4 Create Credential for Authenticated Access to Source System

Create a credential to enable authenticated access to the source system for the purpose of copying or moving a tenant database.

Prerequisites

- You have a user in the system database of the target system with the system privilege CREDENTIAL ADMIN.
- The user in the tenant database of the source system who will be used for access has the system privilege DATABASE ADMIN.

Context

You create a credential in the secure internal credential store of the system database of target system.

The credential store is used in SAP HANA to securely store credentials required for outbound connections. For more information about the secure internal credential store, see the *SAP HANA Security Guide*.

Procedure

In the system database of the target system, create a credential by executing the following SQL statement:

```
CREATE CREDENTIAL FOR COMPONENT 'DATABASE_REPLICATION' PURPOSE
'<host:internal_port_of_system_DB_of_source_system>' TYPE 'PASSWORD' USING
'user="<user_in_system_DB_of_source_system_with_DATABASE_ADMIN>";password="<passw
ord>"'
```

The values required for each parameter are as follows:

Table 20:

Parameter	Required Value
COMPONENT	DATABASE_REPLICATION
PUROPOSE	Host name and internal port number of the system database of the source system
TYPE	PASSWORD
USING	User name of a user in the tenant database of the source system with the system privilege DATABASE ADMIN

Sample Code

```
CREATE CREDENTIAL FOR COMPONENT 'DATABASE_REPLICATION' PURPOSE
'host123456.acme.corp:30001' TYPE 'PASSWORD' USING
'user="DATABASE_ADMINSTRATOR";password="<password>"'
```

Task overview: Preparing to Copy or Move a Tenant Database [page 109]

Previous task: Set Up Trust Relationship Between Target and Source Systems [page 112]

Next task: Back Up Tenant Database [page 114]

7.3.5 Back Up Tenant Database

Back up the tenant database that will be copied or moved.

Context

You can back up the tenant database from the system database or from the tenant database directly. For more information, see *Creating Backups* in the *SAP HANA Administration Guide*.

Task overview: Preparing to Copy or Move a Tenant Database [page 109]

Previous task: Create Credential for Authenticated Access to Source System [page 113]

7.4 Copy a Tenant Database to Another System

Copy a tenant database from one SAP HANA system to another. The new copied tenant database runs as a separate, independent database.

Prerequisites

- All general system prerequisites are fulfilled. For more information, see *Copying and Moving Tenant Databases Between Systems*.
- All preparatory steps have been completed. For more information, see *Preparing to Copy or Move a Tenant Database*.
- You have a user in the system database of the target system with the system privilege `DATABASE ADMIN` and `CATALOG READ`.

Procedure

1. Create a tenant database in the target system as a copy of the original tenant database in the source system.

   You do this by executing the `CREATE DATABASE` statement (for example, in the SQL console of the SAP HANA studio):

   Code Syntax

   ```
   CREATE DATABASE <new_database_name> [AT [LOCATION]
   '<host_name>[:<port_number>[@<source_host_name>:<source_port_number>]]']
   {ADD '<service_type>' [AT [LOCATION]
   '<host_name>[:<port_number>[@<source_host_name>:<source_port_number>]]']}…
   AS REPLICA OF <source_database_name> AT [LOCATION]
   '<host_name>:<port_number>' [OS USER '<user_name>' OS GROUP '<group_name>']
   [NO START]
   ```

   Note

   - As the location of the source tenant database, you specify the host name and port number for internal communication of the **system database** of the source system.
   - If you specify a service list, the number and type of services must match the source database.

Sample Code

```
CREATE DATABASE TARGET_DATABASE AS REPLICA OF SOURCE_DATABASE AT
'host123456.acme.corp:30001';
```

With the execution of this statement, the system database of the target system does the following:

- Establishes a secure connection to the system database of the source system
- Creates a new tenant database with the same topology as the tenant database in the source system
- Starts replicating data between the services in the source tenant database and the corresponding services in the target database

2. Monitor replication progress of data replication from the original tenant database to the new tenant database.

   Use the system view `SYS_DATABASES.M_DATABASE_REPLICAS` to monitor the status of data replication in the system database of the target system or `SYS.M_DATABASES` to monitor directly in the new tenant database.

   The current status of replication is shown in the field `REPLICATION_STATUS`.

   The view `SYS(_DATABASES).M_DATABASE_REPLICAS_STATISTICS` provides detailed information about the replication process at the service level.

   ### Tip

   If the replication status is `ERROR`, use system view `SYS_DATABASES.M_DATABASE_REPLICA_STATISTICS` to investigate further.

   If you cannot create a new replica because the source database is still in status "REPLICATING" even though the target database is already dropped, execute the statement `ALTER DATABASE <database_name> CANCEL REPLICA` on the source system to clean up the system before re-attempting replication.

   ### Note

   You can also monitor from the source system if you opened communication between the systems in both directions. For more information, see *Open Communication From Target to Source System*.

3. When replication status is `ACTIVE` (indicating that the new tenant database is in sync with the original tenant database), stop replication and finalize the copy by executing the following statement in the system database of the target system:

   ```
   ALTER DATABASE <new_database_name> FINALIZE REPLICA
   ```

   With the execution of the above statement, the system database of the target system performs the following actions in the new tenant database:

   - Starts the new tenant database
   - Changes the root key for data volume encryption and re-encrypts data in the new database if data volume encryption is enabled
   - Deletes remote identities of database users if the original tenant database was configured for cross-database access

Next Steps

Perform the required manual post-move tasks.

Related Information

Copying and Moving Tenant Databases Between Systems [page 99]
Preparing to Copy or Move a Tenant Database [page 109]
Perform Manual Post-Copy/Move Tasks [page 119]

7.5 Move a Tenant Database to Another System

Move a tenant database in one SAP HANA system to another. After a move, the original tenant database is deleted and the new tenant database takes over.

Prerequisites

- All general system prerequisites are fulfilled. For more information, see *Copying and Moving Tenant Databases Between Systems*.
- All preparatory steps have been completed. For more information, see *Preparing to Copy or Move a Tenant Database*.
- You have a user in the system database of the target system with the system privilege `DATABASE ADMIN` and `CATALOG READ`.

Procedure

1. Create a tenant database in the target system as a copy of the original tenant database in the source system.

   You do this by executing the `CREATE DATABASE` statement:

   Code Syntax

   ```
   CREATE DATABASE <new_database_name> [AT [LOCATION]
   '<host_name>[:<port_number>[@<target_host_name>:<target_port_number>]]']
   {ADD '<service_type>' [AT [LOCATION]
   '<host_name>[:<port_number>[@<target_host_name>:<target_port_number>]]']}…
   AS REPLICA OF <source_database_name> AT [LOCATION]
   '<host_name>:<port_number>' [OS USER '<user_name>' OS GROUP '<group_name>']
   [NO START]
   ```

Note

- As the location of the source tenant database, you specify the host name and port number for internal communication of the **system database** of the source system.
- If you specify a service list, the number and type of services must match the source database.

Sample Code

```
CREATE DATABASE TARGET_DATABASE AS REPLICA OF SOURCE_DATABASE AT
'host123456.acme.corp:30001';
```

With the execution of this statement, the system database of the target system does the following:

- Establishes a secure connection to the system database of the source system
- Creates a new tenant database with the same topology as the tenant database in the source system
- Starts replicating data between the services in the source tenant database and the corresponding services in the target database

2. Monitor replication progress of data replication from the original tenant database to the new tenant database.

   Use the system view `SYS_DATABASES.M_DATABASE_REPLICAS` to monitor the status of data replication in the system database of the target system or `SYS.M_DATABASES` to monitor directly in the new tenant database.

   The current status of replication is shown in the field `REPLICATION_STATUS`.

   The view `SYS(_DATABASES).M_DATABASE_REPLICAS_STATISTICS` provides detailed information about the replication process at the service level.

   ### Tip

   If the replication status is `ERROR`, use system view `SYS_DATABASES.M_DATABASE_REPLICA_STATISTICS` to investigate further.

   If you cannot create a new replica because the source database is still in status "REPLICATING" even though the target database is already dropped, execute the statement `ALTER DATABASE <database_name> CANCEL REPLICA` on the source system to clean up the system before re-attempting replication.

   ### Note

   You can also monitor from the source system if you opened communication between the systems in both directions. For more information, see *Open Communication From Target to Source System*.

3. When replication status is `ACTIVE` (indicating that the new tenant database is in sync with the original tenant database), stop replication and finalize the move by executing the following statement in the system database of the target system:

   ```
   ALTER DATABASE <new_database_name> FINALIZE REPLICA DROP SOURCE DATABASE
   ```

With the execution of the above statement, the system database of the target system performs the following actions:

- Starts the new tenant database
- Changes the root key for data volume encryption and re-encrypts data in the new database if data volume encryption is enabled
- Drops the original tenant database in the source system

> **Note**
>
> To ensure that the new tenant database can be recovered to the most recent consistent state after the move, data backups are not deleted as part of the move process. This is important in the event that a backup is created in the original tenant database after replication has finished but before the original database is finally deleted.

- Deletes any communication paths configured for cross-database access that reference the original tenant database in the other tenant databases of the source system

Next Steps

Perform the required manual post-move tasks.

Related Information

Copying and Moving Tenant Databases Between Systems [page 99]
Preparing to Copy or Move a Tenant Database [page 109]
Perform Manual Post-Copy/Move Tasks [page 119]

7.6 Perform Manual Post-Copy/Move Tasks

After you have committed the copy or move and the new tenant database is up and running, you must perform several manual tasks.

Procedure

1. Perform a full data backup of the new tenant database (copy only).
2. Reverse the preparatory steps required to secure the copy or move process:
   - Close network communication from the target system to the source system. See *Open Communication From Target to Source System*.

- Delete the certificate collection with purpose DATA REPLICATION created to secure external communication between the systems. See *Set Up Trust Relationship Between Target and Source Systems*.
- Delete the credential used by the system database of the target system to access the source system. See *Create Credential for Authenticated Access to Source System*.

3. If necessary, reconfigure parameters in *.ini files with tenant-database-specific values. See *Configuration Parameters in Multiple-Container Systems*.
4. Reconfigure cross-database access, if required. See *Enable and Configure Cross-Database Access*.

Related Information

Open Communication From Target to Source System [page 111]
Set Up Trust Relationship Between Target and Source Systems [page 112]
Create Credential for Authenticated Access to Source System [page 113]
Enable and Configure Cross-Database Access [page 54]
Configuration Parameters in Multiple-Container Systems [page 15]

7.7 Tutorial: Moving a Tenant Database

In this tutorial, we'll move an existing tenant database from a test system to a production system.

Scenario

M01 is a test system. A newly developed application is running here in the tenant database NEW_APP_TEST. After several rounds of successful application testing, it is time to move the application to the production system, M02. The production system already has some applications running in other tenant databases. Since we would like to have full database isolation between applications, the new application should also run in a separate tenant database in the production system. We will therefore move the tenant database NEW_APP_TEST in system M01 to system M02, where it will be called NEW_APP_PROD.

Tools

To do the move, we will use several tools:

- Command line tool
- SAP HANA studio
- SAP Web Dispatcher Administration
- SAP HANA cockpit

Authorization

We will need the following authorizations in both systems:

- Credentials of operating system users m01adm and m02adm
- Database user in the system database with the following system privileges and roles:
  - `CATALOG READ`
  - `INIFILE ADMIN`
  - `SSL ADMIN`
  - `DATABASE ADMIN`
  - `BACKUP ADMIN`
  - `sap.hana.xs.wdisp.admin::WebDispatcherAdmin`
  - `sap.hana.security.cockpit.roles::EditCertificateStore`
  - `sap.hana.security.cockpit.roles::MaintainCertificateCollections`
  - `sap.hana.security.cockpit.roles::MaintainCertificates`

As the target system is running in high isolation mode, you will also need root user access to be able to create an operating group and user for the new tenant database.

Steps

This tutorial takes you through the following steps:

1. Verify TLS/SSL Configuration of Internal Communication Channels in M01 and M02 [page 122]
   Verify that TLS/SSL is enabled on internal communication channels in source system M01 and target system M02. M02 is running in high isolation mode, so this will automatically be the case.
2. Open Communication From M02 to M01 [page 123]
   Open communication from the target system M02 to the source system M01 by enabling services in the source system to listen on all network interfaces.
3. Configure TLS/SSL for External Communication in M01 and M02 [page 124]
   Configure the external communication channel for TLS/SSL in the source system M01 and the target system M02 using in-database certificate collections.
4. Set Up Trust Relationship Between M02 and M01 [page 127]
   Create an in-database certificate collection in the system database of the target system M02 and add the public-key certificate of the source system M01. During the move, this certificate will later be used to secure communication between the systems via external SQL connections.
5. Create Credential for Authenticated Access from M02 to M01 [page 130]
   Create a credential to enable authenticated access to the source system M01 from target system M02 for the purpose of moving the tenant database.
6. Back Up Tenant Database NEW_APP_TEST [page 131]
   Back up the tenant database NEW_APP_TEST in the source system.
7. Create OS Group and OS for New Tenant Database in M02 [page 132]
   Create an operating system group and user for the new tenant database that you will create in target system M02. This is necessary because M02 is configured for high isolation.

8. Move Tenant Database NEW_APP_TEST [page 132]
   Move the tenant database NEW_APP_TEST in the source system M01 to the target system M02 as the database NEW_APP_PROD. After the move, NEW_APP_TEST is deleted and NEW_APP_PROD is up and running.

7.7.1 Verify TLS/SSL Configuration of Internal Communication Channels in M01 and M02

Verify that TLS/SSL is enabled on internal communication channels in source system M01 and target system M02. M02 is running in high isolation mode, so this will automatically be the case.

Procedure

In the system database of the source system (M01):

1. In the SAP HANA studio, open the *Configuration* tab of the Administration editor.
2. Navigate to the `[communication]` section of the `global.ini` file.
3. Verify that value of the parameter `ssl` is set to **`systemPKI`** at the `SYSTEM` layer.

Name	Default	System
global.ini		
[] communication		
ssl	off	systemPKI

> **Note**
>
> If **`systemPKI`** is not set, set it and restart the system.

In the system database of the target system (M02):

4. Navigate to the `[communication]` section of the `global.ini` file.
5. In the SAP HANA studio, open the *Configuration* tab of the Administration editor.
6. Verify that value of the parameter `database_isolation` is set to **`high`** at the `SYSTEM` layer.

Name	Default	System
global.ini		
[] multidb		
database_isolation	low	high

Task overview: Tutorial: Moving a Tenant Database [page 120]

Next task: Open Communication From M02 to M01 [page 123]

7.7.2 Open Communication From M02 to M01

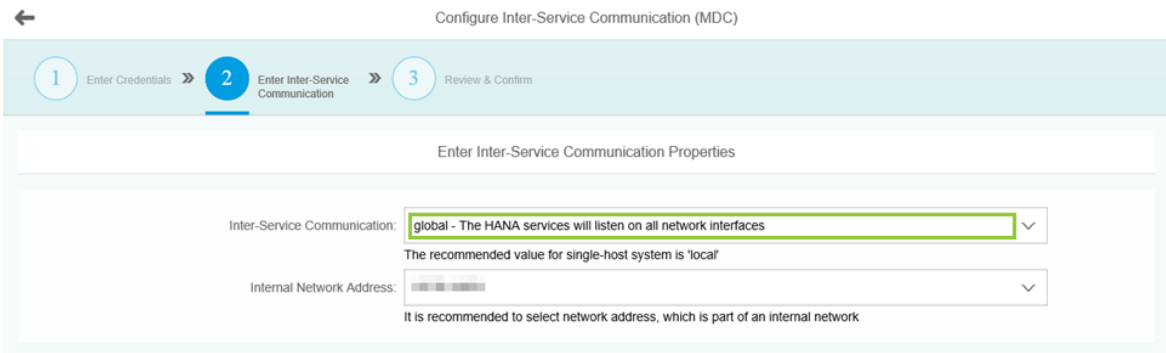
Open communication from the target system M02 to the source system M01 by enabling services in the source system to listen on all network interfaces.

Procedure

In the system database of the source system (M01):

1. Open the SAP HANA database lifecycle manager by entering the following URL in a browser:
   `https://<host>:1129/lmsl/HDBLCM/M01/index.html`
2. Click the tile *Configure Inter-Service Communication*.

3. When prompted, enter the password of the m01adm user.
4. Select the setting *global*.



> **Note**
>
> You do not need to change any other settings.

5. Click *Run* to apply the new setting.
6. Close the application and log out.

In the system database of the target system (M02):

7. Repeat the steps above.

Task overview: Tutorial: Moving a Tenant Database [page 120]

Previous task: Verify TLS/SSL Configuration of Internal Communication Channels in M01 and M02 [page 122]

Next task: Configure TLS/SSL for External Communication in M01 and M02 [page 124]

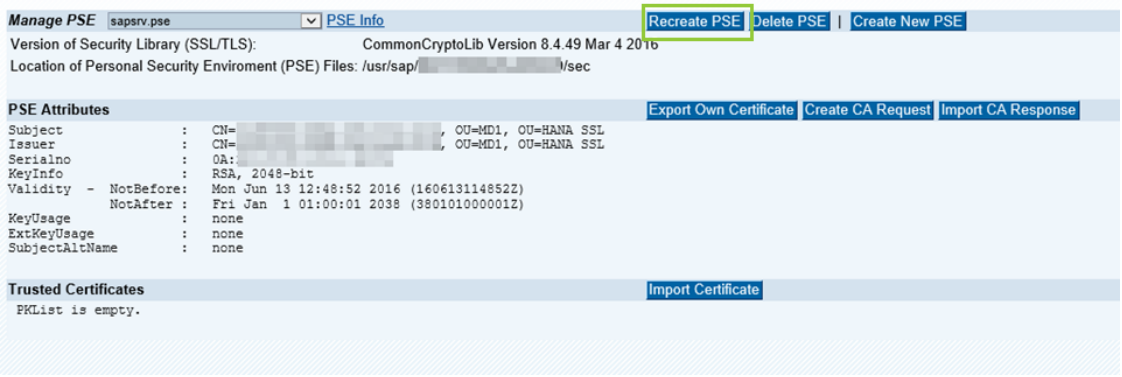
7.7.3 Configure TLS/SSL for External Communication in M01 and M02

Configure the external communication channel for TLS/SSL in the source system M01 and the target system M02 using in-database certificate collections.

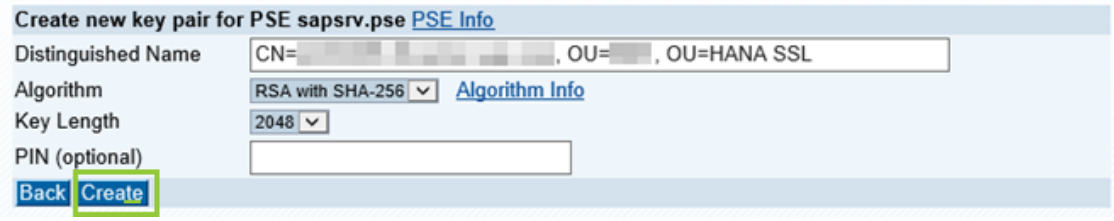
Procedure

In the system database of the source system (M01):

1. Generate a new public and private key pair for the system.
   a. Open the SAP Web Dispatcher Administration tool by entering the following URL in your browser:
      `http://<host_FQDN>:80<instance>/sap/hana/xs/wdisp/admin`
   b. Navigate to the *PSE Management* screen and select the PSE `sapsrv.pse`.
   c. Choose *Recreate PSE*.



   d. Verify the information and choose *Create*.



2. Extract the system's own certificate, certificate chain and private key from the PSE file `sapsrv.pse`:
   a. Download the script `extract_certificates.py` from SAP Note 2175664 to the server.

b. From the command line, log on to the server as m01adm and navigate to the directory containing the PSE file `sapsrv.pse`:

```
cd /usr/sap/M01/HDB<instance>/mdchost/sec
```

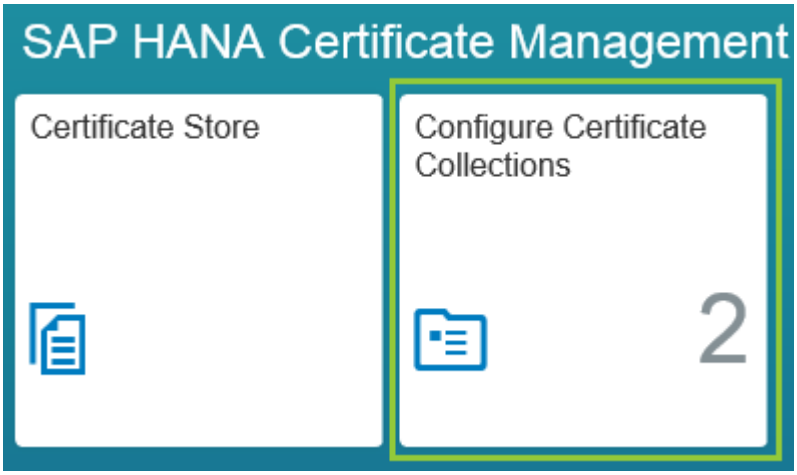
c. Execute the script as follows:

```
python <path_to_script> -p sapsrv.pse
```

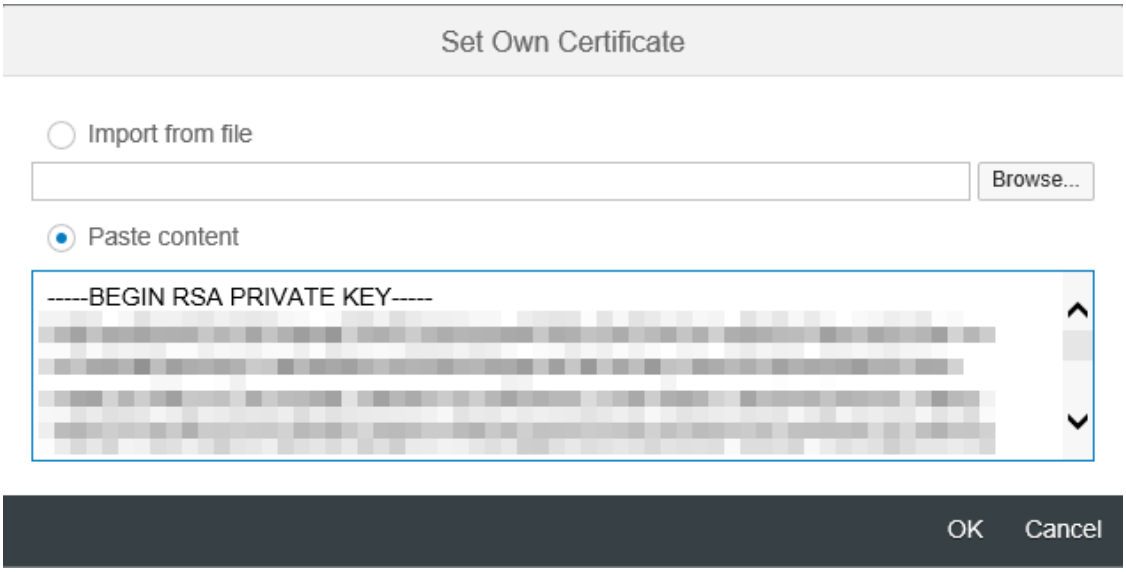
d. Copy the output starting `-----BEGIN RSA PRIVATE KEY-----` and ending `-----END CERTIFICATE-----` to a text file.

> **Caution**
>
> The extraction process prints the private key directly in PEM form to the screen. Make sure that no unauthorized access is possible to this key information because this could compromise the security of the own certificate and the private key.

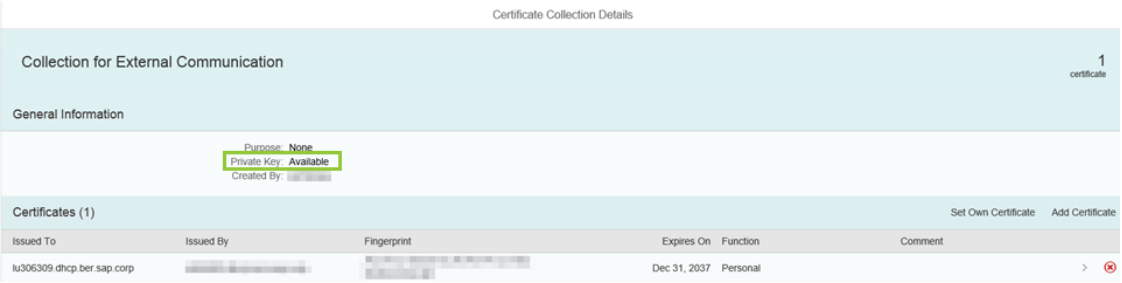
3. Create a certificate collection for external communication:
   a. Open the SAP HANA cockpit by entering the following URL in your browser:
      `http://<host_FQDN>:80<instance>/sap/hana/admin/cockpit`
   b. Open the *Configure Certificate Collections* app:



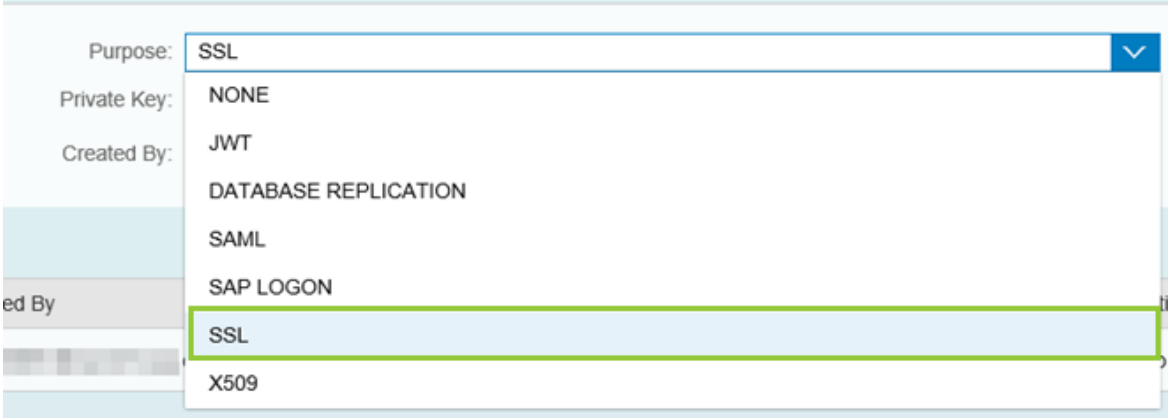
   c. Create a new collection.
   d. In the new collection, click *Set Own Certificate* and paste in the extracted certificate contents:



Notice that it is now indicated that the collection has a private key available:



4. Set the purpose of the collection to *SSL*:



5. Save the collection and exit the application.

In the system database of the target system:

6. Repeat the steps above.

Task overview: Tutorial: Moving a Tenant Database [page 120]

Previous task: Open Communication From M02 to M01 [page 123]

Next task: Set Up Trust Relationship Between M02 and M01 [page 127]

7.7.4 Set Up Trust Relationship Between M02 and M01

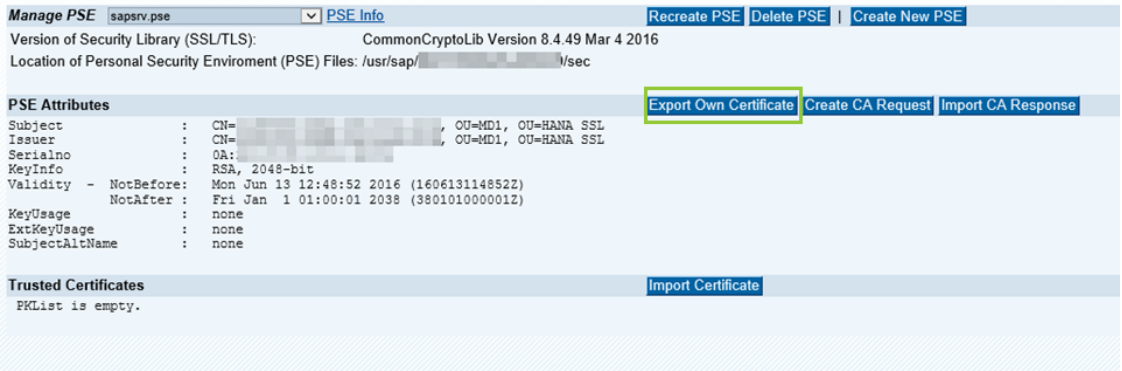
Create an in-database certificate collection in the system database of the target system M02 and add the public-key certificate of the source system M01. During the move, this certificate will later be used to secure communication between the systems via external SQL connections.

Procedure

In the system database of the source system (M01):

1. Get the public-key certificate of the source system.
   a. Open the SAP Web Dispatcher Administration tool by entering the following URL in your browser:

      `http://<host_FQDN>:80<instance>/sap/hana/xs/wdisp/admin`
   b. Navigate to the *PSE Management* screen and select the PSE `sapsrv.pse`.
   c. Click *Export Own Certificate*.



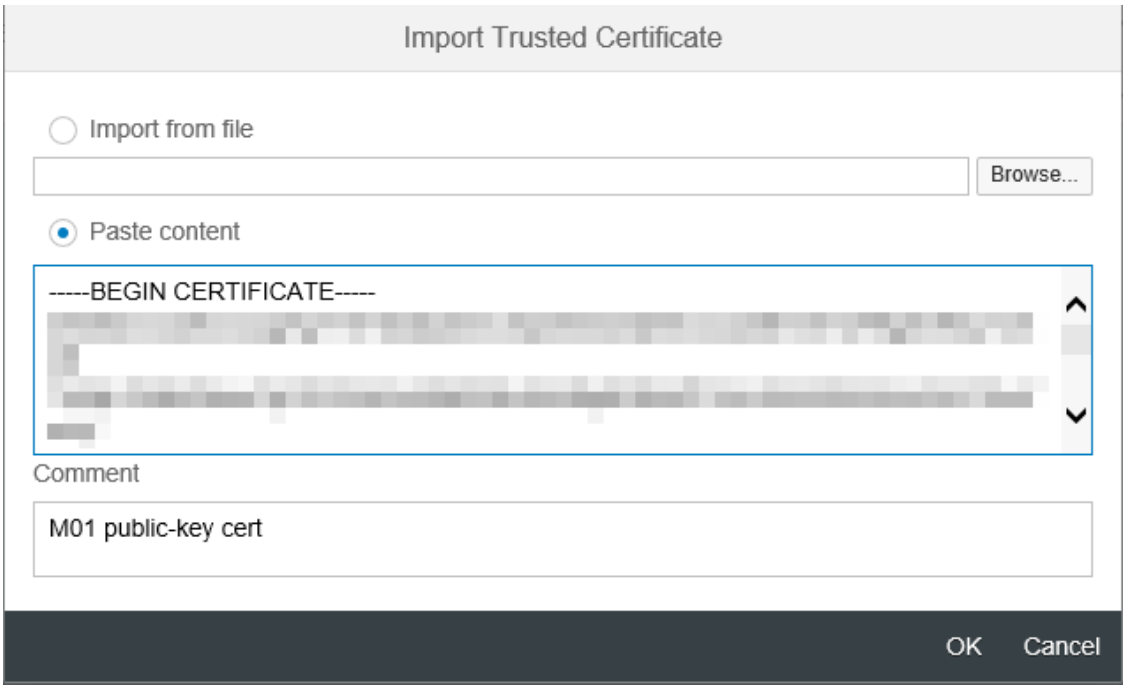
   d. Copy the certificate contents to a text file.

In the system database of the target system (M02):

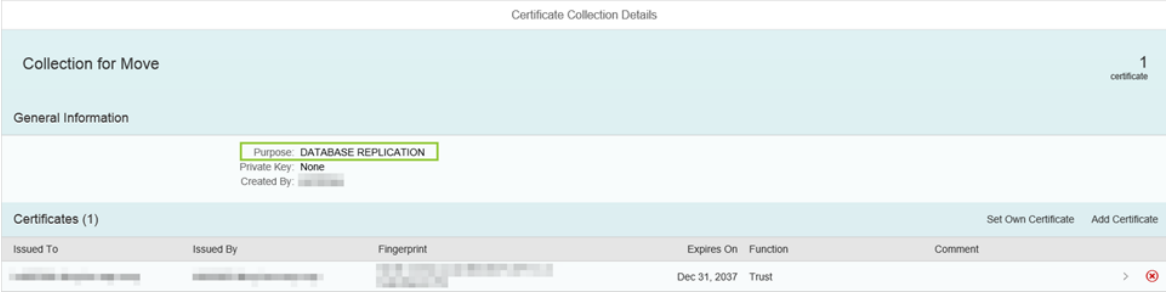
2. Import the public-key certificate of the source system into the certificate store:
   a. Open the SAP HANA cockpit by entering the following URL in your browser:

      `http://<host_FQDN>:80<instance>/sap/hana/admin/cockpit`
   b. Open the *Certificate Store* app.

c. Click *Import* and paste the contents of the source system certificate.



3. Create a certificate collection for the purpose of database replication.
   a. Open the *Configure Certificate Collections* app.
   b. Create a new collection.
   c. Click *Add Certificate* and select the public-key certificate of the source system.
4. Set the purpose of the collection to *DATABASE REPLICATION* and save.



Task overview: Tutorial: Moving a Tenant Database [page 120]

Previous task: Configure TLS/SSL for External Communication in M01 and M02 [page 124]

Next task: Create Credential for Authenticated Access from M02 to M01 [page 130]

7.7.5 Create Credential for Authenticated Access from M02 to M01

Create a credential to enable authenticated access to the source system M01 from target system M02 for the purpose of moving the tenant database.

Procedure

In the system database of the target system (M02):

1. In the SAP HANA studio, open the SQL console.
2. Execute the following statement:

```
CREATE CREDENTIAL FOR COMPONENT 'DATABASE_REPLICATION' PURPOSE
'<host>:<internal_port>' TYPE 'PASSWORD' USING
'user="<database_administrator_user>";password="<password>"'
```

> **Note**
>
> As the PUROPOSE you enter the host name and internal port number of the system database of the source system M01. The user to be specified after the keyword USING is a user in the system database of the source system M01 with the system privilege DATABASE ADMIN.

Task overview: Tutorial: Moving a Tenant Database [page 120]

Previous task: Set Up Trust Relationship Between M02 and M01 [page 127]

Next task: Back Up Tenant Database NEW_APP_TEST [page 131]

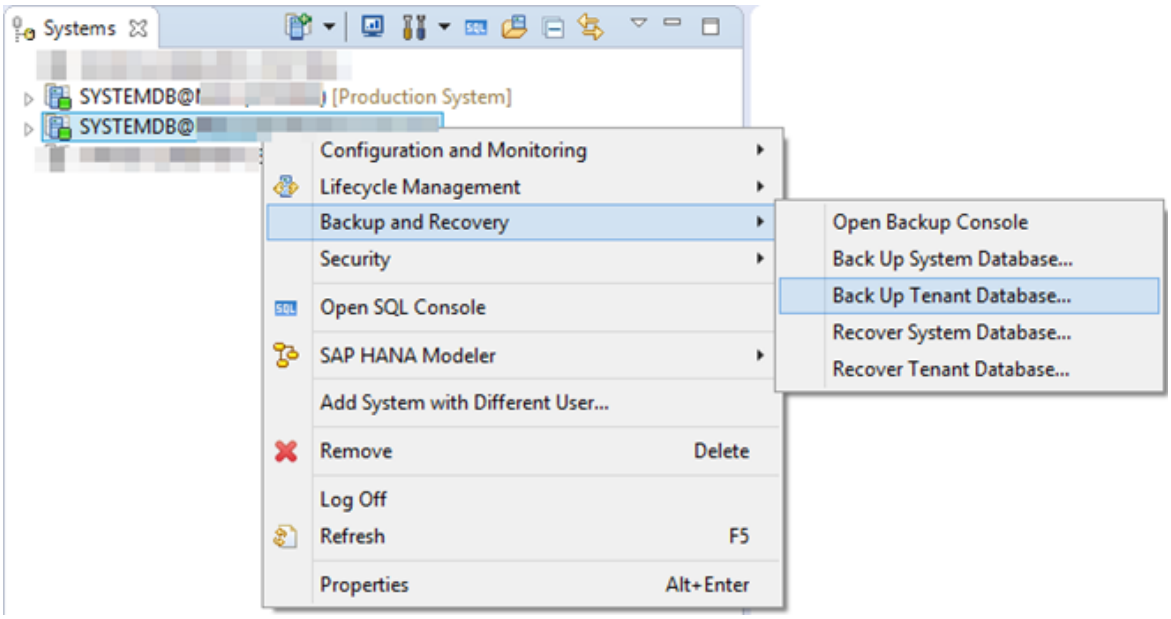
7.7.6 Back Up Tenant Database NEW_APP_TEST

Back up the tenant database NEW_APP_TEST in the source system.

Procedure

In the system database of the source system (M01):

1. In the SAP HANA studio, right-click the source system and choose *Backup and Recovery* > *Backup Tenant Database*.



2. Select the tenant database NEW_APP_TEST.
3. Select *Complete Data Backup*.
4. Specify the destination type and location for the backup.
5. Specify the backup prefix.
6. Choose *Next* and *Finish*.

Task overview: Tutorial: Moving a Tenant Database [page 120]

Previous task: Create Credential for Authenticated Access from M02 to M01 [page 130]

Next task: Create OS Group and OS for New Tenant Database in M02 [page 132]

7.7.7 Create OS Group and OS for New Tenant Database in M02

Create an operating system group and user for the new tenant database that you will create in target system M02. This is necessary because M02 is configured for high isolation.

Procedure

In the target system (M02):

1. As root user, log on to the server on which the name server of the system database is running.
2. Create a new group:

   ```
   groupadd new_app_prod
   ```

3. Create a new user, specifying `sapsys` as the primary group:

   ```
   useradd -g sapsys new_app_prod
   ```

4. Add the new user to the `m02shm` group and their own group as secondary groups:

   ```
   usermod -G m02shm,new_app_prod new_app_prod
   ```

Task overview: Tutorial: Moving a Tenant Database [page 120]

Previous task: Back Up Tenant Database NEW_APP_TEST [page 131]

Next task: Move Tenant Database NEW_APP_TEST [page 132]

7.7.8 Move Tenant Database NEW_APP_TEST

Move the tenant database NEW_APP_TEST in the source system M01 to the target system M02 as the database NEW_APP_PROD. After the move, NEW_APP_TEST is deleted and NEW_APP_PROD is up and running.

Procedure

In the system database of the target system (M02):

1. In the SAP HANA studio, open the SQL console.

2. Create the new tenant database NEW_APP_PROD in the target system as a copy of the original tenant database NEW_APP_TEST in the source system.

```
CREATE DATABASE NEW_APP_PROD AS REPLICA OF NEW_APP_TEST AT
'<host_name>:<port_number>' OS USER 'new_app_prod' OS GROUP 'new_app_prod'
```

3. Monitor replication progress of data replication from the NEW_APP_TEST to NEW_APP_PROD.

```
SELECT * FROM "SYS"."M_DATABASE_REPLICAS"
```

4. When replication status is `ACTIVE`, stop replication and finalize the move by executing the following statement:

```
ALTER DATABASE NEW_APP_PROD FINALIZE REPLICA DROP SOURCE DATABASE
```

Task overview: Tutorial: Moving a Tenant Database [page 120]

Previous task: Create OS Group and OS for New Tenant Database in M02 [page 132]

8 Using SAP Web Dispatcher for Load Balancing with Tenant Databases

If an SAP HANA system has multiple instances of SAP HANA extended services, classic model (SAP HANA XS classic) and is distributed across multiple hosts, you can implement an external SAP Web Dispatcher to distribute the load of inbound HTTP requests and to enure high availability. This is also possible for systems with tenant databases, but requires additional configuration.

SAP Note 1855097 describes how to configure an external SAP Web Dispatcher for single-container SAP HANA systems. The following sections describe the additional configuration required for systems with tenant databases.

Before You Start

Note the following points:

- The external SAP Web Dispatcher is a separate installation and does not form part of the SAP HANA system. For use with multiple-container systems, it must have a minimum version of **745 Patch Level 21**.
- An SAP Web Dispatcher process also runs on all SAP HANA hosts on which an instance of SAP HANA XS is active. This internal SAP Web Dispatcher is a fixed part of the SAP HANA system. In a system with tenant databases, this internal SAP Web Dispatcher must also be configured to enable HTTP access to individual databases. For more information, see *Configure HTTP(S) Access to Multitenant Database Containers*.
- All information and configuration steps described in SAP Note 1855097 are still valid. In particular, the parameter `wdisp/filter_xs_internal_uri` has to be set to **`false`** in the `webdispatcher.ini` configuration file of your SAP HANA system.
- The configuration described in the following sections describes access to tenant databases. However, it is also valid for the system database. For the Web Dispatcher, there is no difference between tenant databases and the system database.
- The SAP Web Dispatcher handles only HTTP(S) access to SAP HANA.
- For more information about configuring secure HTTPS access, see *Configure HTTP(S) Access to Multitenant Database Containers* (internal Web Dispatcher configuration) and *Configuring SAP Web Dispatcher to Support SSL* in the SAP HANA Web Dispatcher documentation.

Related Information

SAP Note 1855097

Configure HTTP(S) Access to Multitenant Database Containers via SAP HANA XS Classic [page 62]

Configuring SAP Web Dispatcher to Support SSL

Virtual-Host-Based Routing [page 135]

Configuring an External SAP Web Dispatcher for Tenant Databases [page 137]

8.1 Virtual-Host-Based Routing

An example explains the basics of virtual-host-based routing.

The Website `travelapp.com` provides Web-based services for information about popular travel destinations. Services are implemented as separate applications, which run on separate Web servers on one host (`travelapp.com`). Virtual host names are used to distinguish between the available services: `london.travelapp.com` and `paris.travelapp.com`. Both virtual host names are aliases for `travelapp.com`. This can be illustrated as follows:

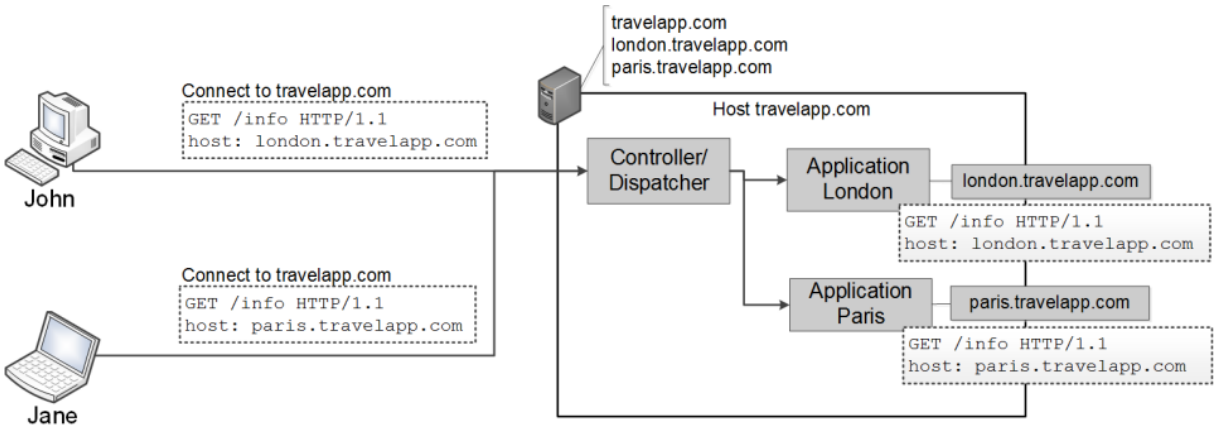


Figure 9: Virtual-Host-Based Routing

John wants to read information about London. Therefore, he enters `london.travelapp.com` into his browser. As `london.travelapp.com` is an alias for `travelapp.com`, the browser sends the HTTP request to `travelapp.com`, but it uses `london.travelapp.com` as the host header of this request. The request arrives at a controller or dispatcher process on `travelapp.com`. This dispatcher process decides whether to forward the request to the Web server responsible for displaying information about London or the Web server responsible for displaying information about Paris. This decision is made based on the host header, that is the host name that the user originally entered into the browser. `london.travelapp.com` is assigned to the application for London and `paris.travelapp.com` is assigned to the application for Paris.

Jane requires information about Paris and enters `paris.travelapp.com` into her browser. This request also arrives at the dispatcher process and is dispatched to the Paris application based on the host header of the request.

Load Balancing

`travelapp.com` has proved to be a successful service with many users. As a result, one host is no longer sufficient, and the application has been installed on a second host. In addition, a load balancer is needed to distribute requests between the two hosts. The aliases `london.travelapp.com` and

`paris.travelapp.com` have to be changed to point to the host of the load balancer to guarantee that all requests are handled by the load balancer (`dmz.travelapp.com`). This can be illustrated as follows:

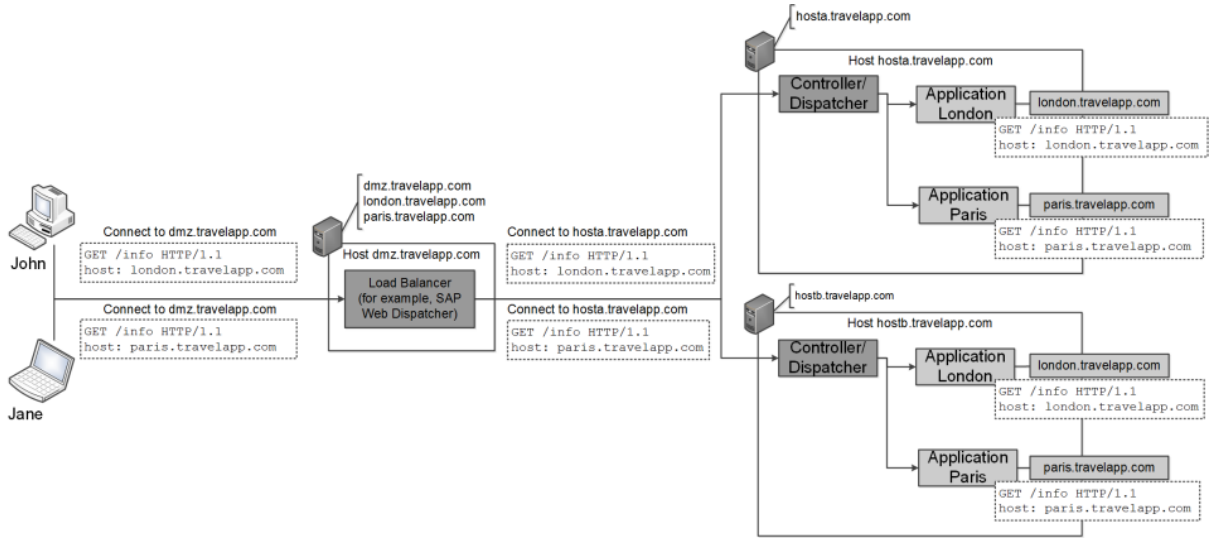


Figure 10: Virtual-Host-Based Routing with Load Balancing

John again wants to read information about London. Therefore, he enters `london.travelapp.com` into his browser. As `london.travelapp.com` is an alias for `dmz.travelapp.com`, the browser sends the HTTP request to `dmz.travelapp.com`, but it uses `london.travelapp.com` as the host header of this request. This request arrives at the load balancer, which simply forwards the request to `hosta.travelapp.com` or `hostb.travelapp.com` based on the current load. It must not change the host header of the request because this request is later necessary in the dispatcher. After that, the dispatcher handles the request as if no load balancer is involved, regardless of the fact that the host name in the host header actually points to another host.

SAP HANA Multitenant Database Containers

Translated to the context of SAP HANA multitenant database containers, the load balancer is an external SAP Web Dispatcher, and the dispatcher is the system-internal SAP Web Dispatcher, as illustrated in the following figure:

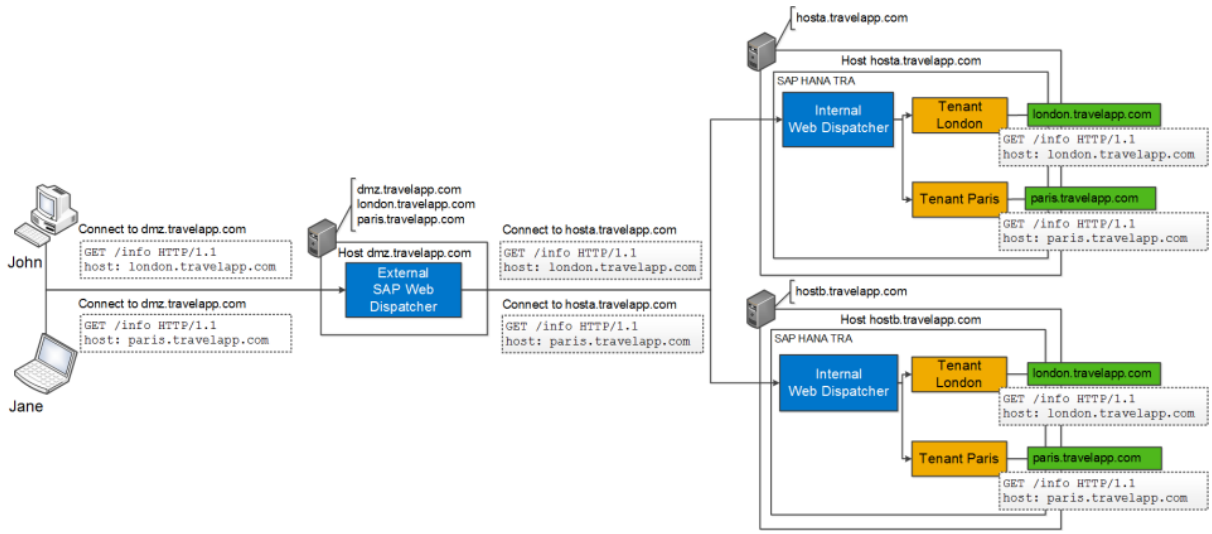


Figure 11: Virtual-Host-Based Routing for SAP HANA Multitenant Database Containers

8.2 Configuring an External SAP Web Dispatcher for Tenant Databases

Virtual host names for differentiated HTTP access to tenant databases are configured in the system-internal SAP Web Dispatcher. If you're using an external SAP Web Dispatcher for load balancing, you must also configure the external Web Dispatcher. Otherwise, information about the selected virtual hosts can't be transported to the SAP HANA system.

> **Remember**
>
> All of the configuration settings mentioned here are done in the **external** Web Dispatcher and not in the internal Web Dispatcher that is part of the SAP HANA system. The external Web Dispatcher is a separate installation and does not form part of the SAP HANA system. Before you can configure the external Web Dispatcher, the internal Web Dispatcher must already have been configured to enable HTTP access to individual databases. For more information, see *Configure HTTP(S) Access to Multitenant Database Containers*.

Single Versus Multiple Tenant Access via External Web Dispatcher

Every tenant database that needs to be accessed through the external Web Dispatcher (and your system is running SPS 11 or higher) requires a `wdisp/system_<XX>` parameter entry in the external Web Dispatcher

profile (`sapwebdisp.pfl`). The `XSSRV` subparameter specifies the XS server to connect to, and the `XSVHOST` subparameter specifies the virtual host name of the tenant database.

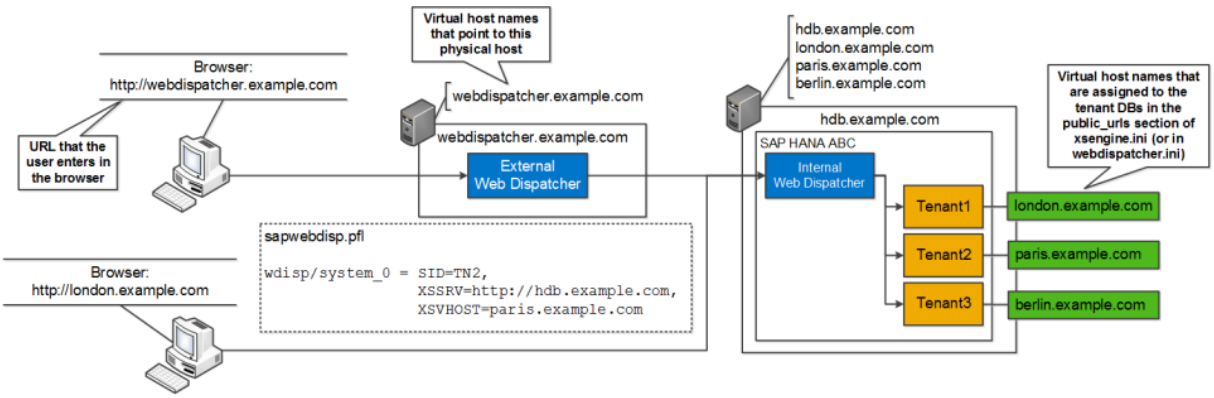
Example

```
wdisp/system_<xx> = SID=<3-digit ID>, XSSRV=http://<physical host name of SAP
HANA server>:<port>, XSVHOST=<virtual host name of the tenant>
```

Note

Virtual host names are configured in the `public_urls section` of the `xsengine.ini` configuration file (or in `webdispatcher.ini`). The is part of the **internal** Web Dispatcher configuration. For more information, see *Configure HTTP(S) Access to Multitenant Database Containers*.

If only **one tenant database** needs to be accessed through the external Web Dispatcher (and your system is running SPS 11 or higher), a single `wdisp/system_<XX>` entry for the tenant database with the above configuration is sufficient, as depicted in the following figure:



Note

The figure shows a simplified depiction of the Web Dispatcher profile (`sapwebdisp.pfl`). In the real configuration, the `XSSRV` subparameter requires port numbers, and line breaks are not allowed.

Figure 12: Access to a Single Tenant Database (SPS 11 and higher)

Note

An external Web Dispatcher is not mandatory to access a single tenant. But there are scenarios in which an external Web Dispatcher is required, for example sophisticated applications (for example, some SAP Fiori scenarios) or for security purposes. For more information, see the relevant application documentation and the SAP Web Dispatcher documentation.

If **all or multiple tenant databases** need to be accessed (and your system is running SPS 09 or SPS 10), in addition to a `wdisp/system_<XX>` parameter entry for each tenant database, it is necessary to configure the external Web Dispatcher to differentiate between the various tenant databases. Since each `wdisp/system_<xx>` entry in the external Web Dispatcher represents one tenant database, incoming HTTP requests have to be mapped to a tenant database, that is to the right `wdisp/system_<xx>` entry in the internal Web Dispatcher configuration.

Virtual hosts names are used to configure tenant differentiation. Two scenarios are possible:

- **Option 1**: Tenant databases are accessed via HTTP through the external Web Dispatcher only; there is no direct HTTP access to the tenant databases (recommended)
- **Option 2**: Tenants databases are accessed via HTTP both through the external Web Dispatcher and directly, bypassing the external Web Dispatcher
  This configuration requires additional virtual host names and is more complex than option 1. However, this option is useful if the external Web Dispatcher is being added to an existing landscape.

Related Information

SAP Web Dispatcher Documentation on SAP Help Portal
Configure HTTP(S) Access to Multitenant Database Containers via SAP HANA XS Classic [page 62]
Option 1: Configuring Access to Multiple (or All) Tenant Databases Through External Web Dispatcher Only [page 139]
Option 2: Configuring Access to Multiple (or All) Tenant Databases Through External Web Dispatcher and Directly [page 142]

8.2.1 Option 1: Configuring Access to Multiple (or All) Tenant Databases Through External Web Dispatcher Only

Use this configuration if you want tenant databases to be accessed through the external Web Dispatcher only. With this configuration, there is no direct HTTP access to the tenant databases.

The main part of this configuration involves setting the virtual host names of tenant databases configured in the external Web Dispatcher profile to point to the host of the external Web Dispatcher, instead of the host of the SAP HANA system. As a result, all requests to the virtual host name of a tenant database first go to the external Web Dispatcher and are then forwarded to the internal Web Dispatcher in the SAP HANA system.

> **Remember**
>
> Before you can configure the external Web Dispatcher, the internal Web Dispatcher must already have been configured to enable HTTP access to individual databases. For more information, see *Configure HTTP(S) Access to Multitenant Database Containers*.

Single-Host Systems

The `wdisp/system_<xx>` entry for each tenant database is configured in the external Web Dispatcher profile as follows:

- `XSSRV` specifies the actual physical SAP HANA host name and port to which to requests are sent.
- `XSVHOST` specifies the virtual host name of the tenant database to which requests are sent.

> **Note**
>
> If a tenant database has multiple virtual host names assigned, only one needs to be entered in `XSVHOST`.

- `SRCVHOST` specifies the virtual host name that is used to map incoming HTTP requests to the `wdisp/system` entry that represents a particular tenant.

> **Note**
>
> With this configuration option, `XSVHOST` and `SRCVHOST` are always identical.

The following figure depicts this configuration in a single-host system:

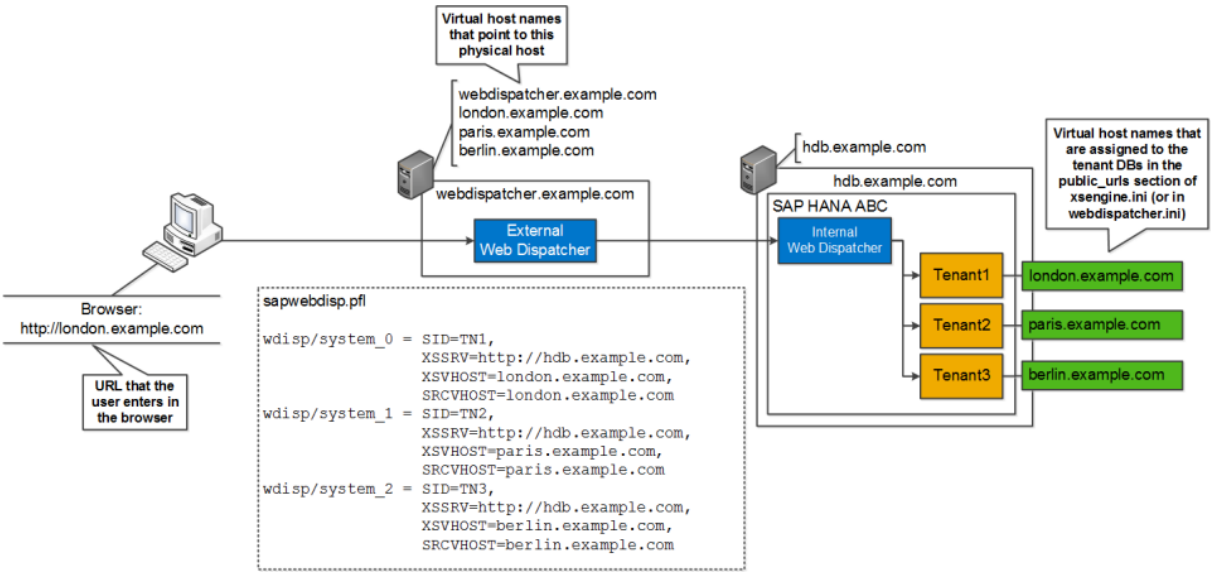


Figure 13: Access to Multiple Tenant Databases (Single Host)

> **Note**
>
> The figure shows a simplified depiction of the Web Dispatcher profile (`sapwebdisp.pfl`). In the real configuration, the `XSSRV` subparameter requires port numbers, and line breaks are not allowed.

In the example depicted above, what happens when the user enters **`london.example.com`** into her browser?

1. The browser opens a TCP/IP connection to `webdispatcher.example.com` because `london.example.com` is only an alias name for `webdispatcher.example.com`.
2. The browser sends an HTTP request over this connection. The host header of this HTTP request is `london.example.com`, which is the URL that the user entered.
3. The external Web Dispatcher receives the HTTP request, checks the host header and uses this to map the request to a `wdisp/system` entry. As `london.example.com` is the `SRCVHOST` value for `wdisp/system_0`, the request is associated with `wdisp/system_0`.
4. The external Web Dispatcher opens a TCP/IP connection to the `XSSRV` value of `wdisp/system_0` (**`hdb.example.com`**).

5. The external Web Dispatcher sets the destination of the request to the tenant database specified in the `XSVHOST` subparameter of `wdisp/system_0` (**`london.example.com`**) by injecting a proprietary HTTP header into the request.
6. The internal SAP HANA Web Dispatcher receives the request. Because of the injected HTTP header field, it identifies that the request is destined for tenant database 1 and forwards it to the XS server of tenant database 1.

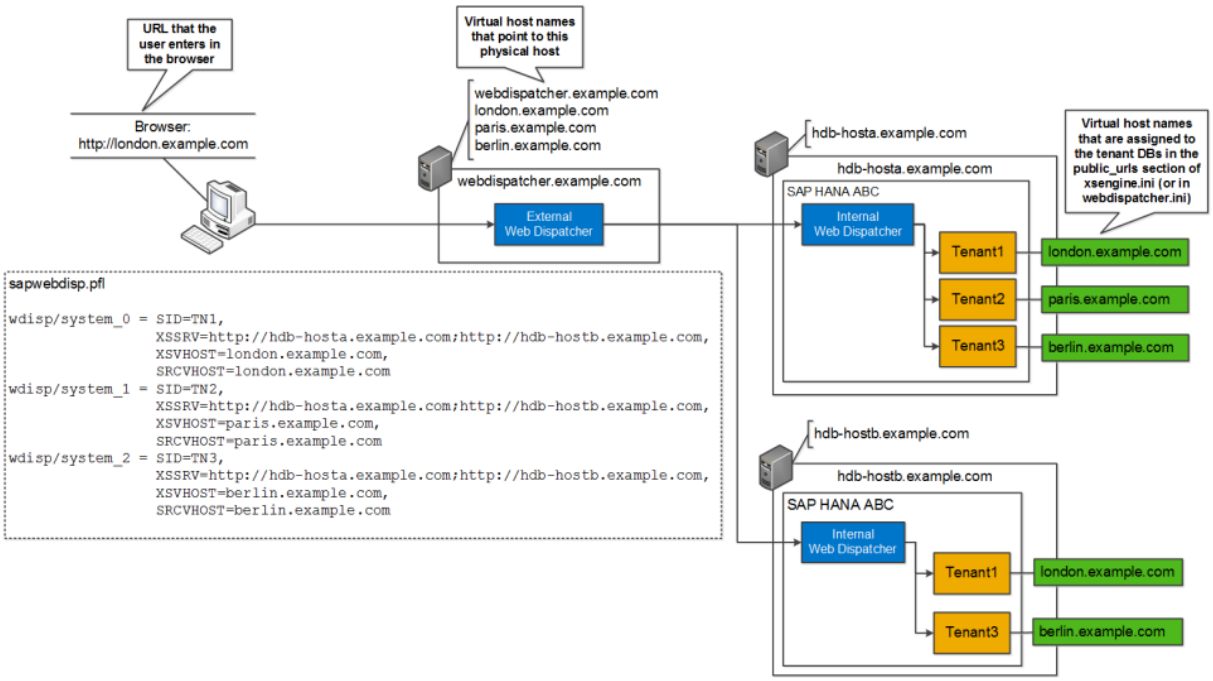
> **Note**
>
> The injected HTTP header field can only be used in systems running SPS 11 or higher. Because of this, additional manual configuration is required for SPS 9 and SPS 10. See *Additional Configuration Required in SPS 09 and SPS 10*.

Multiple-Host Systems

In a multiple-host system, the external Web Dispatcher must be configured to connect to all hosts. This means that all hosts with a running XS server (or that may have an XS server in the future) have to be entered as the value for `XSSRV` as a semi-colon (;) separated list. Even if a tenant database is not running on a host, you should add the host to the list anyway. This will enable the smooth moving of tenant databases without the need to change the external Web Dispatcher configuration.

> **Remember**
>
> In the internal Web Dispatcher configuration, the virtual host name in the `XSVHOST` subparameter must be to assigned to the tenant database on **all** hosts.

The following figure depicts the configuration in a multiple-host system:



> **Note**
>
> The figure shows a simplified depiction of the Web Dispatcher profile (`sapwebdisp.pfl`). In the real configuration, the `XSSRV` subparameter requires port numbers, and line breaks are not allowed.

Figure 14: Access to Multiple Tenant Databases (Multiple Hosts)

Now what happens when the user enters **`london.example.com`** into her browser?

The process is identical to the single-host scenario with one exception: The external Web Dispatcher periodically checks which host(s) a tenant database is actually running on. If a tenant database is running on multiple hosts, the external Web Dispatcher performs load balancing between these hosts.

Related Information

Additional Configuration Required in SPS 09 and SPS 10 [page 145]
Configure HTTP(S) Access to Multitenant Database Containers via SAP HANA XS Classic [page 62]

8.2.2 Option 2: Configuring Access to Multiple (or All) Tenant Databases Through External Web Dispatcher and Directly

Use this configuration if you want tenant databases to be accessed both through the external Web Dispatcher and directly, bypassing the external Web Dispatcher.

With this configuration, additional virtual host names are required for each tenant database. These virtual host names point to the physical host name of the external Web Dispatcher. The virtual host names that are assigned to the tenant databases still point to the host of the SAP HANA system.

> **Remember**
>
> Before you can configure the external Web Dispatcher, the internal Web Dispatcher must already have been configured to enable HTTP access to individual databases. For more information, see *Configure HTTP(S) Access to Multitenant Database Containers*.

Single-Host Systems

The `wdisp/system_<xx>` entry for each tenant database is then configured in the external Web Dispatcher profile as follows:

- `XSSRV` specifies the actual physical host name and port to which to requests are sent.
- `XSVHOST` specifies the virtual host name of the tenant database to which requests are sent.

> **Note**
>
> If a tenant database has multiple virtual host names assigned, only one needs to be entered in `XSVHOST`.

- `SRCVHOST` specifies the virtual host name that is used to map incoming HTTP requests to the `wdisp/system_<xx>` that represents a particular tenant.

The following figure depicts this configuration in a single-host system:

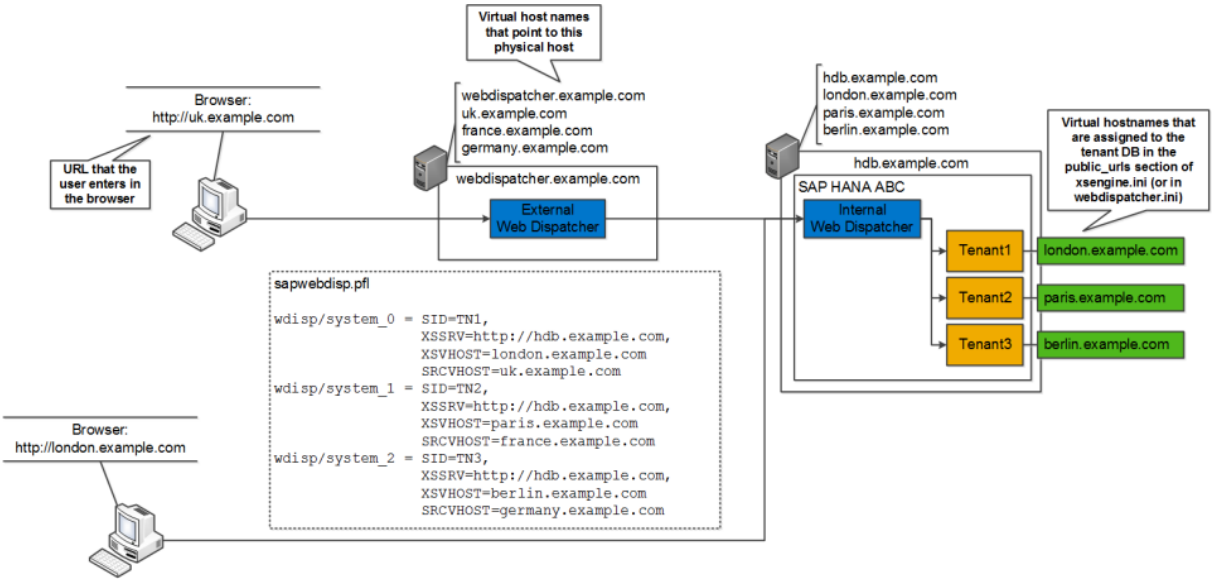


Figure 15: Access to Multiple Tenant Databases in Single-Host System (SPS 11 and Higher)

> **Note**
>
> The figure shows a simplified depiction of the Web Dispatcher profile (`sapwebdisp.pfl`). In the real configuration, the `XSSRV` subparameter requires port numbers, and line breaks are not allowed.

In the example depicted above, what happens when the user enters **`uk.example.com`** into his browser?

1. The browser opens a TCP/IP connection to `webdispatcher.example.com` because `uk.example.com` is only an alias name for `webdispatcher.example.com`.
2. The browser sends an HTTP request over this connection. The host header of this HTTP request is `uk.example.com`, which is the URL that the user entered.
3. The external Web Dispatcher receives the HTTP request, checks the host header and uses it to map the request to a `wdisp/system` entry. As `uk.example.com` is the `SRCVHOST` value for `wdisp/system_0`, the request is associated with `wdisp/system_0`.
4. The external Web Dispatcher opens a TCP/IP connection to the `XSSRV` value of `wdisp/system_0` (`hdb.example.com`).
5. The external Web Dispatcher sets the destination of the request to the tenant database specified in the `XSVHOST` parameter of `wdisp/system_0` (`london.example.com`) by injecting a proprietary HTTP header into the request.
6. The internal SAP HANA Web Dispatcher receives the request. Because of the injected HTTP header field, it identifies that the request is destined for tenant database 1 and forwards it to the XS server of tenant database 1.

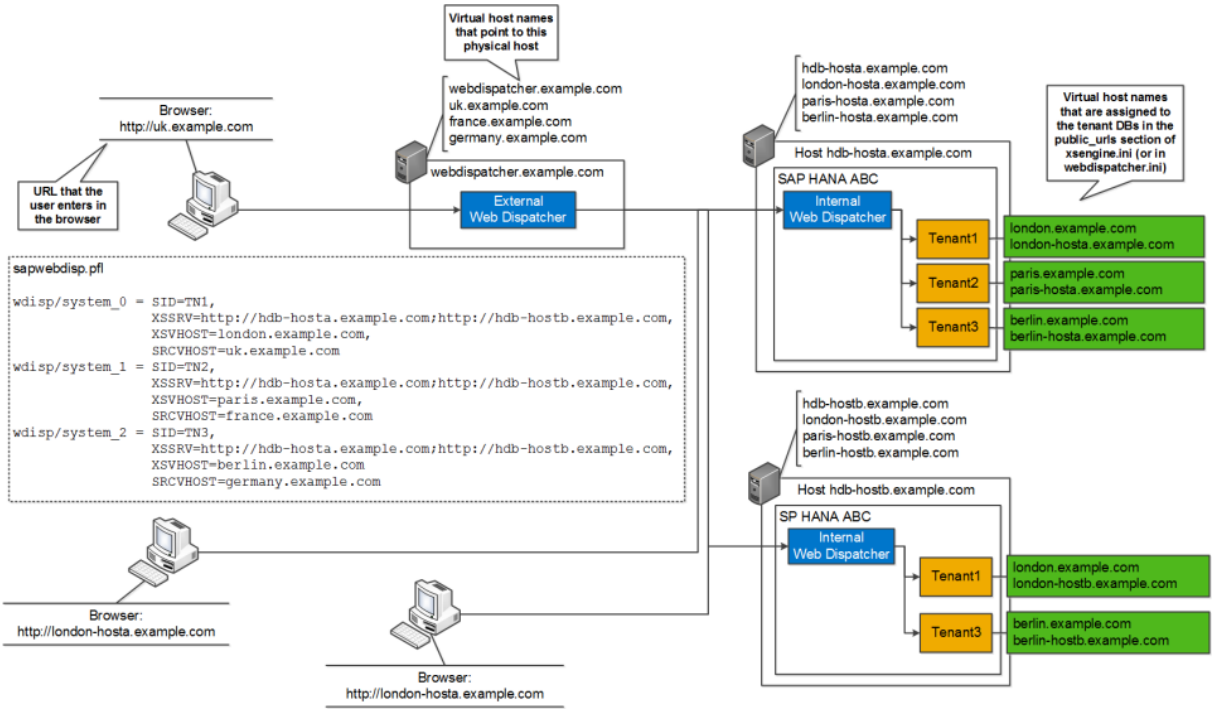
> **Note**
>
> The injected HTTP header field can only be used in SPS 11 or higher. Because of this additional manual configuration is required for SPS 9 and SPS 10. See *Additional Configuration Required in SPS 09 and SPS 10*.

Multiple-Host Systems

In a multiple-host system, the external Web Dispatcher must be configured to connect to all hosts. This means that all hosts with a running XS server (or that may have an XS server in the future) have to be entered as the value for `XSSRV` as a semi-colon (;) separated list. Even if a tenant database is not running on a host, you should add the host to the list anyway. This will enable the smooth moving of tenant databases without the need to change the external Web Dispatcher configuration.

> **Remember**
>
> In the internal Web Dispatcher configuration, the virtual host name in the `XSVHOST` subparameter must be to assigned to the tenant on **all** hosts.

The following figure depicts the configuration in a multiple-host system:



> **Note**
>
> The figure shows a simplified depiction of the Web Dispatcher profile (`sapwebdisp.pfl`). In the real configuration, the `XSSRV` subparameter requires port numbers, and line breaks are not allowed.

Figure 16: Access to Multiple Tenant Databases in Multiple-Host System (SPS 11 and Higher)

What happens after the user enters **`uk.example.com`** into his browser?

The process is identical to the single-host scenario with one exception: The external Web Dispatcher periodically checks on which host(s) a tenant database is actually running. If a tenant database is running on multiple hosts, the external Web Dispatcher performs load balancing between these hosts.

Related Information

Additional Configuration Required in SPS 09 and SPS 10 [page 145]
Configure HTTP(S) Access to Multitenant Database Containers via SAP HANA XS Classic [page 62]

8.2.3 Additional Configuration Required in SPS 09 and SPS 10

In SAP HANA systems running SPS 09 and SPS 10, the configurations described in the previous sections are valid. However, an additional step is required for some access scenarios.

Additional configuration is required for the following access scenarios:

- Single tenant database access through an external Web Dispatcher (access through the external Web Dispatcher and directly)
- Multiple tenant database access through an external SAP Web Dispatcher and directly (that is, configuration option 2 for multiple tenant database access)

Additional configuration is not required for access to multiple tenant databases through an external Web Dispatcher only (that is, configuration option 1 for multiple tenant database access).

> **Note**
>
> This additional configuration does not have to be reverted after you upgrade from SPS 09 or SPS 10 to SPS 11 or higher.

Access to Single Tenant Database

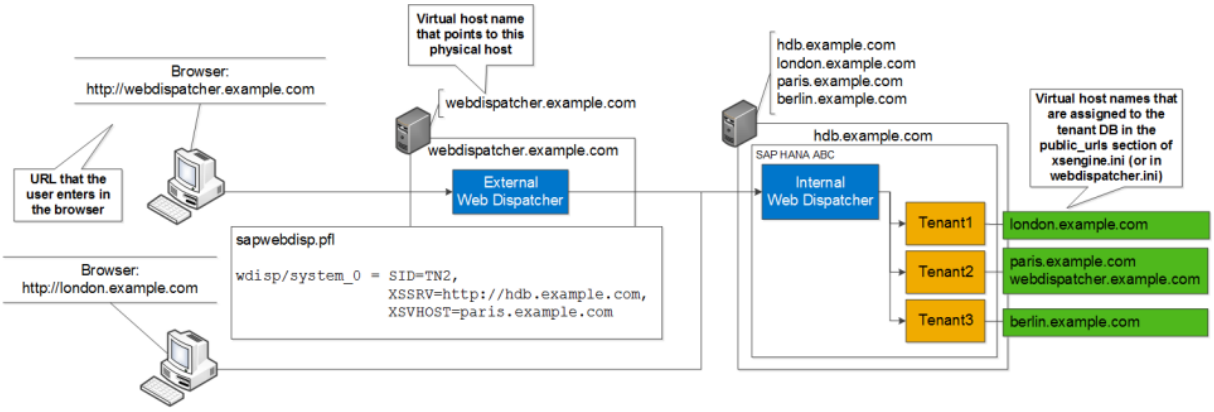
For access to a single tenant database, the host name of the external Web Dispatcher must be added to the `SRCVHOST` subparameter of the `wdisp/system` parameter for the tenant database in the `webdispatcher.ini` configuration file.

In SPS 09, you do this by editing the `webdispatcher.ini` file directly.

Example

ALTER SYSTEM ALTER CONFIGURATION ('webdispatcher.ini', 'system, '<tenant_DB_name>') SET (wdisp/system_xx) = ' SID=<3-character ID>, EXTSRV=http://localhost:3$(SAPSYSTEM)<internal xsengine port>, SRCVHOST=<Tenant DB FQDN>;<host name of external Web Dispatcher>' WITH RECONFIGURE;

In **SPS 10**, you add the host name to the `public_urls` section of the `xsengine.ini` configuration file, for example, `http_webdispatcher_url=http://<hostname>:80$(SAPSYSTEM)`.



Note

The figure shows a simplified depiction of the Web Dispatcher profile (`sapwebdisp.pfl`). In the real configuration, the `XSSRV` subparameter requires port numbers, and line breaks are not allowed. Also note that in SPS 09, the internal Web Dispatcher configuration must be done directly in the `webdispatcher.ini` file. Configuration using the `public_urls` section of the `xsengine.ini` file was introduced with SPS 10.

Figure 17: Access to a Single Tenant Database (SPS 09 and 10)

Access to Multiple Tenant Databases

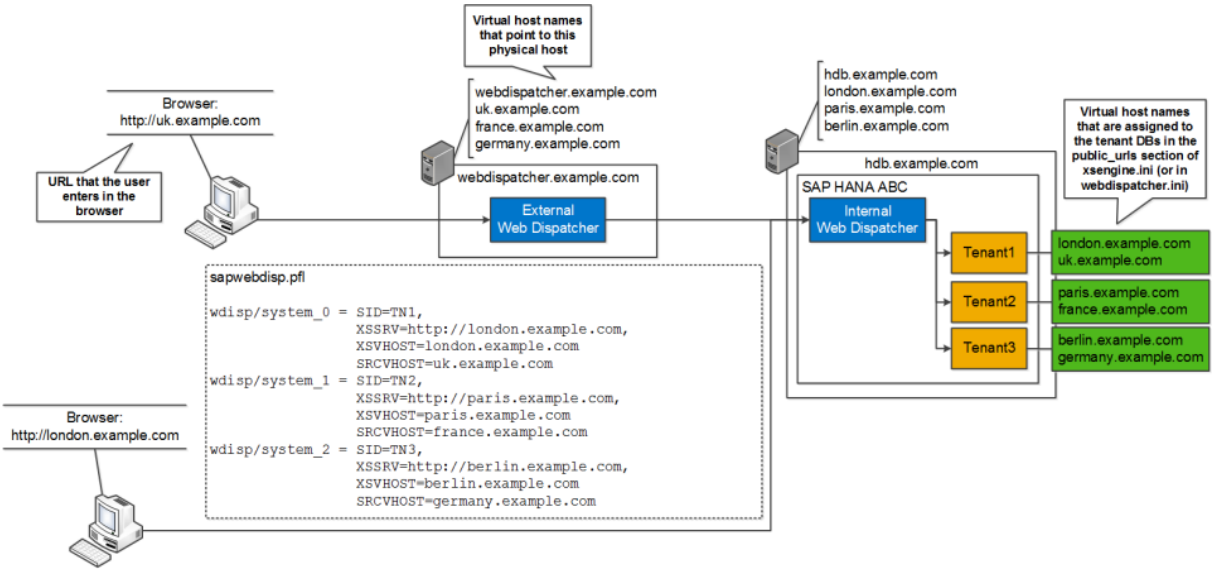
For access to multiple tenant databases, the additional virtual host names of the tenant databases configured in the external Web Dispatcher profile (that is the host name specified in the `SRCVHOST` subparameter of the `wdisp/system` parameter for a tenant database) must be added to the `SRCVHOST` subparameter of the `wdisp/system` parameter for each tenant databases in the `webdispatcher.ini` configuration file.

In SPS 09, you do this by editing the `webdispatcher.ini` file directly:

Example

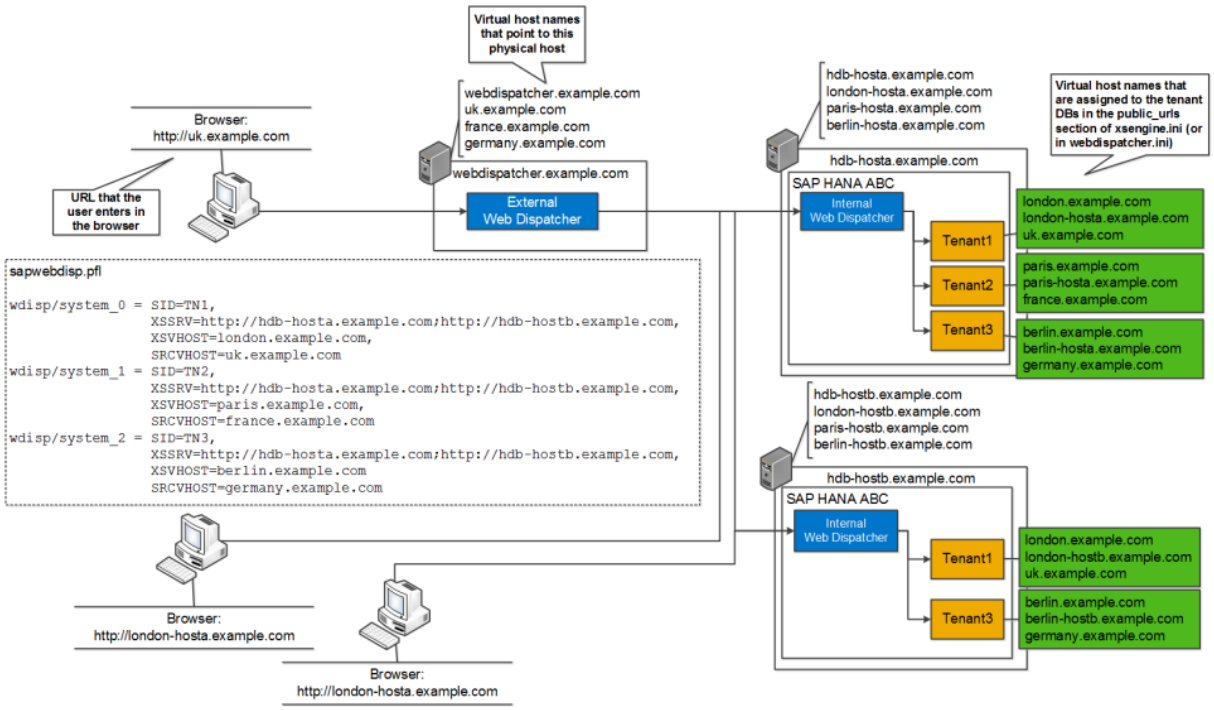
ALTER SYSTEM ALTER CONFIGURATION ('webdispatcher.ini', 'system, '<tenant_DB_name>') SET (wdisp/system_xx) = ' SID=<3-character ID>, EXTSRV=http://localhost:3$(SAPSYSTEM)<internal xsengine port>, SRCVHOST=<Tenant DB FQDN>;<additional virtual host name>' WITH RECONFIGURE;

In **SPS 10**, you add the host name to the `public_urls` section of the `xsengine.ini` configuration file, for example, `http_webdispatcher_url=http://<hostname>:80$(SAPSYSTEM)`.



> **Note**
>
> The figure shows a simplified depiction of the Web Dispatcher profile (`sapwebdisp.pfl`). In the real configuration, the `XSSRV` subparameter requires port numbers, and line breaks are not allowed. Also note that in SPS 09, the internal Web Dispatcher configuration must be done directly in the `webdispatcher.ini` file. Configuration using the `public_urls` section of the `xsengine.ini` file was introduced with SPS 10.

Figure 18: Access to Multiple Tenant Databases in Single-Host System (SPS 09 and 10)



> **Note**
>
> The figure shows a simplified depiction of the Web Dispatcher profile (`sapwebdisp.pfl`). In the real configuration, the `XSSRV` subparameter requires port numbers, and line breaks are not allowed. Also note that in SPS 09, the internal Web Dispatcher configuration must be done directly in the `webdispatcher.ini` file. Configuration using the `public_urls` section of the `xsengine.ini` file was introduced with SPS 10.

Figure 19: Access to Multiple Tenant Databases in Multiple-Host System (SPS 09 and 10)

9 SAP HANA System Replication with Multitenant Database Containers

The usual SAP HANA system replication principles apply for multitenant database containers.

Before you begin preparing a replication strategy for an SAP HANA multiple-container system, you should be aware of the following important points.

- SAP HANA multiple-container systems can only be replicated as the whole system. This means that the system database and all tenant databases are part of the system replication. A takeover can only be performed as a whole system. A takeover on the level of a single tenant database is not possible.
- If a new tenant database is created in a configured SAP HANA system replication, it must be backed up to participate in the replication. Afterwards, the initial data shipping is started automatically for this tenant database. If a takeover is done while the initial data shipping is running and not finished, this new tenant database will not be operational after takeover and will have to be recovered with backup and recovery (see the *SAP HANA Database Backup and Recovery* section of the SAP HANA Administration Guide).
- If a tenant database is recovered on the primary system, the replication will be in an inconsistent state with the secondary site until this tenant database is reinitialized (for more information see – `sr_initialize` in the section *System Replication Command Line Reference* of the SAP HANA Administration).
- If an active tenant database is stopped in a running SAP HANA system replication, it is stopped on the secondary site as well. If a takeover is done while tenant databases (which were part of the system replication) are stopped, they will be in the same state after takeover as they were on the primary site when they were stopped. The tenant databases must be started to complete the takeover.
- If SAP HANA system replication runs in replication mode SYNC with the full sync option enabled, and if the connection to the secondary site is interrupted, no write operations on the primary site are possible. The operation of creating a tenant database, for example, will wait until the connection to the secondary is reestablished or the SQL statement times out.
- With SAP HANA multiple-container systems, the services needed are generated automatically in the tenant databases.
- For SAP HANA system replication, a port offset value of 100 is configured to reserve ports for system replication communication. The port number of the replication port is calculated by adding the value for this replication port offset to the internal port number of the corresponding service. Thus, although the same `<instance number>` is used for primary and secondary systems, the `<instance number>`+1 is reserved for both systems, because this port range is needed for system replication communication. For SAP HANA multiple-container systems, this port offset is set to 10000 shifting the ports from the 3`<instance number>`00 to the 4`<instance number>`00 port range for the services. This is necessary in SAP HANA system replication with SAP HANA multiple-container systems, because after 3`<instance number>`99 is reached new tenant databases allocate port numbers of the next higher instance number.

> **Note**
>
> To avoid interference with ephemeral ports it might be necessary to adjust the OS port range when using SAP HANA system replication in combination with SAP HANA multitenant database containers. On Linux this can be accomplished with the following command in the system startup script: `echo "50000 65535" > /proc/sys/net/ipv4/ip_local_port_range`.

- It is possible to copy or move tenant databases between SAP HANA multiple-container systems using system replication technology. However, you can only use this feature if system replication is not enabled for high availability purposes on either the source or target system for the entire duration of the copy or move process. For more information, see *Copying and Moving Tenant Databases Between Systems*.
- For SAP HANA multiple-container systems running with the HIGH isolation level, the system PKI SSFS data and key file must be copied from the primary system to the same location on the secondary system(s). For more information, see *Increase the System Isolation Level* in the SAP HANA Administration Guide.

For more information on the individual points, see the *Availability and Scalability* section of the SAP HANA Administration Guide.

Related Information

Copying and Moving Tenant Databases Between Systems [page 99]
Increase the System Isolation Level [page 46]
SAP HANA Administration Guide

10 Backing Up and Recovering Multitenant Database Containers

The usual SAP HANA backup and recovery principles apply for multitenant database containers.

Before you begin preparing a backup strategy for an SAP HANA multiple-container system, you should be aware of the following important points. For more information about the individual points, see the *SAP HANA Database Backup and Recovery* section of the *SAP HANA Administration Guide*.

Backup with SAP HANA Multitenant Database Containers

- The system database can initiate backups of both the system database itself and of individual tenant databases.
  A tenant database can create its own backups without using the system database.
  This option is enabled by default.
- The backup location in the file system is specified system-wide. Backups of tenant databases are always created in sub-directories of this location.
- SAP HANA backups created with release 1.0 SPS10 or newer can be used to recover to SAP HANA 2.0. This is true for both SAP HANA single-container systems and tenant databases.
- SAP HANA supports high isolation scenarios for third-party tools with SAP HANA multitenant database containers.
  More information: *Isolation Level High for Backups With SAP HANA Multitenant Database Containers and Third-Party Backup Tools* in *Related Information*
- SAP HANA cockpit cannot be used to administer backup and recovery for SAP HANA multitenant database containers.

Recovery with SAP HANA Multitenant Database Containers

- The recovery of a tenant database is always initiated from the system database.
  If tenant databases need to be recovered, they are recovered individually, and not all together in one single operation.
- The system database is recovered in the same way as an SAP HANA single-container system.
  The system database only needs to be recovered if it is corrupted. If only a tenant database is corrupted, the system database does not need to be recovered.

> **Note**
>
> If the system database is shut down for recovery, its tenant databases are automatically shut down as well. This means that, until the recovery of the system database is completed, all its tenant databases are not available.

While a tenant database is being recovered, the system database and any other tenant databases remain online.

- To recover a complete SAP HANA multitenant database container system, the system database needs to be recovered first, and then all the tenant databases are recovered individually.
- When an SAP HANA multitenant database container is recovered, the services needed are generated automatically in the tenant databases.
- With SAP HANA 2.0, an SAP HANA single-container system can be recovered to a tenant database.
- SAP HANA backups created with release 1.0 SPS10 or newer can be used to recover to SAP HANA 2.0. This is true for both SAP HANA single-container systems and tenant databases.
  More information: SAP Note 2096000 *SAP HANA multitenant database containers - Additional Information* in *Related Information*
- With SAP HANA multitenant database containers, recovery using storage snapshots is currently not supported.
- A backup of an SAP HANA multitenant database container can only be recovered to an SAP HANA multitenant database container.
  Tenant database copy using backup and recovery with third-party tools is currently not supported.
  A backup of a tenant database can be recovered to a different SAP HANA multitenant database container using file system-based backups.
- Adding or removing a service breaks the backup history. After you add or remove a service, you must immediately create a full backup in order to be able to recover the database.
- SAP HANA cockpit cannot be used to administer backup and recovery for SAP HANA multitenant database containers.

Related Information

SAP Note 2096000
SAP HANA Administration Guide

11 Important Disclaimer for Features in SAP HANA Platform, Options and Capabilities

SAP HANA server software and tools can be used for several SAP HANA platform and options scenarios as well as the respective capabilities used in these scenarios. The availability of these is based on the available SAP HANA licenses and the SAP HANA landscape, including the type and version of the back-end systems the SAP HANA administration and development tools are connected to. There are several types of licenses available for SAP HANA. Depending on your SAP HANA installation license type, some of the features and tools described in the SAP HANA platform documentation may only be available in the SAP HANA options and capabilities, which may be released independently of an SAP HANA Platform Support Package Stack (SPS). Although various features included in SAP HANA options and capabilities are cited in the SAP HANA platform documentation, each SAP HANA edition governs the options and capabilities available. Based on this, customers do not necessarily have the right to use features included in SAP HANA options and capabilities. For customers to whom these license restrictions apply, the use of features included in SAP HANA options and capabilities in a production system requires purchasing the corresponding software license(s) from SAP. The documentation for the SAP HANA optional components is available in SAP Help Portal at http://help.sap.com/hana_options. If you have additional questions about what your particular license provides, or wish to discuss licensing features available in SAP HANA options, please contact your SAP account team representative.

Important Disclaimers and Legal Information

Coding Samples

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, unless damages were caused by SAP intentionally or by SAP's gross negligence.

Accessibility

The information contained in the SAP documentation represents SAP's current view of accessibility criteria as of the date of publication; it is in no way intended to be a binding guideline on how to ensure accessibility of software products. SAP in particular disclaims any liability in relation to this document. This disclaimer, however, does not apply in cases of willful misconduct or gross negligence of SAP. Furthermore, this document does not result in any direct or indirect contractual obligations of SAP.

Gender-Neutral Language

As far as possible, SAP documentation is gender neutral. Depending on the context, the reader is addressed directly with "you", or a gender-neutral noun (such as "sales person" or "working days") is used. If when referring to members of both sexes, however, the third-person singular cannot be avoided or a gender-neutral noun does not exist, SAP reserves the right to use the masculine form of the noun and pronoun. This is to ensure that the documentation remains comprehensible.

Internet Hyperlinks

The SAP documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. SAP does not warrant the availability and correctness of this related information or the ability of this information to serve a particular purpose. SAP shall not be liable for any damages caused by the use of related information unless damages have been caused by SAP's gross negligence or willful misconduct. All links are categorized for transparency (see: http://help.sap.com/disclaimer).

go.sap.com/registration/contact.html

© 2016 SAP SE or an SAP affiliate company. All rights reserved.
No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.
Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.
These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.
SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.
Please see http://www.sap.com/corporate-en/legal/copyright/index.epx for additional trademark information and notices.