HA201

SAP HANA 2.0 SPS05 - High Availability and Disaster Tolerance Administration

PARTICIPANT HANDBOOK
INSTRUCTOR-LED TRAINING

Course Version: 17
Course Duration: 3 Day(s)
e-book Duration: 7 Hours 5 Minutes
Material Number: 50155431

SAP Copyrights, Trademarks and Disclaimers

Typographic Conventions

American English is the standard used in this handbook.

The following typographic conventions are also used.

This information is displayed in the instructor's presentation	
Demonstration	
Procedure	
Warning or Caution	
Hint	
Related or Additional Information	
Facilitated Discussion	
User interface control	Example text
Window title	Example text

Contents

Course Overview

TARGET AUDIENCE
This course is intended for the following audiences:

- Technology Consultant
- Database Administrator
- System Administrator

UNIT 1

SAP HANA High Availability Features Overview

UNIT OBJECTIVES

- Understand the different SAP HANA high availability features
- Describe the disaster recovery features in SAP HANA
- Explain the fault recovery features of SAP HANA

Unit 1
Lesson 1

Explaining the SAP HANA High Availability Features

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand the different SAP HANA high availability features

SAP HANA High Availability

Business Example

For your company's SAP ERP and SAP Business Warehouse (SAP BW) systems, high availability and disaster tolerance are important requirements that need to be built into the landscape architecture.

Your SAP ERP and SAP BW systems are running on the SAP HANA database, which is why you are looking into the native high availability and disaster tolerance features of the SAP HANA database system. You want to learn how to incorporate these features into your company's landscape architecture.

SAP HANA and High Availability

SAP HANA is specifically developed to take full advantage of the capabilities provided by modern hardware to increase application performance. By keeping all relevant data in the main memory, data processing operations are significantly accelerated.

Another core design principle for SAP HANA is scalability. The SAP HANA database can be distributed across multiple hosts to achieve scalability in terms of size and user concurrency. A distributed (also called "scale-out") SAP HANA system spreads the data efficiently over the available servers, thereby achieving high scaling without I/O delays.

For a company, the loss of critical business systems directly translates into loss of revenue. In almost every company, this is unacceptable. Therefore, the goal of every company should be business continuity, and consequently they should use systems designed for continuous operation even in the presence of inevitable failures. These mission-critical systems require high availability that is built-in on every level of the landscape, and should not rely on external tools to provide these features.

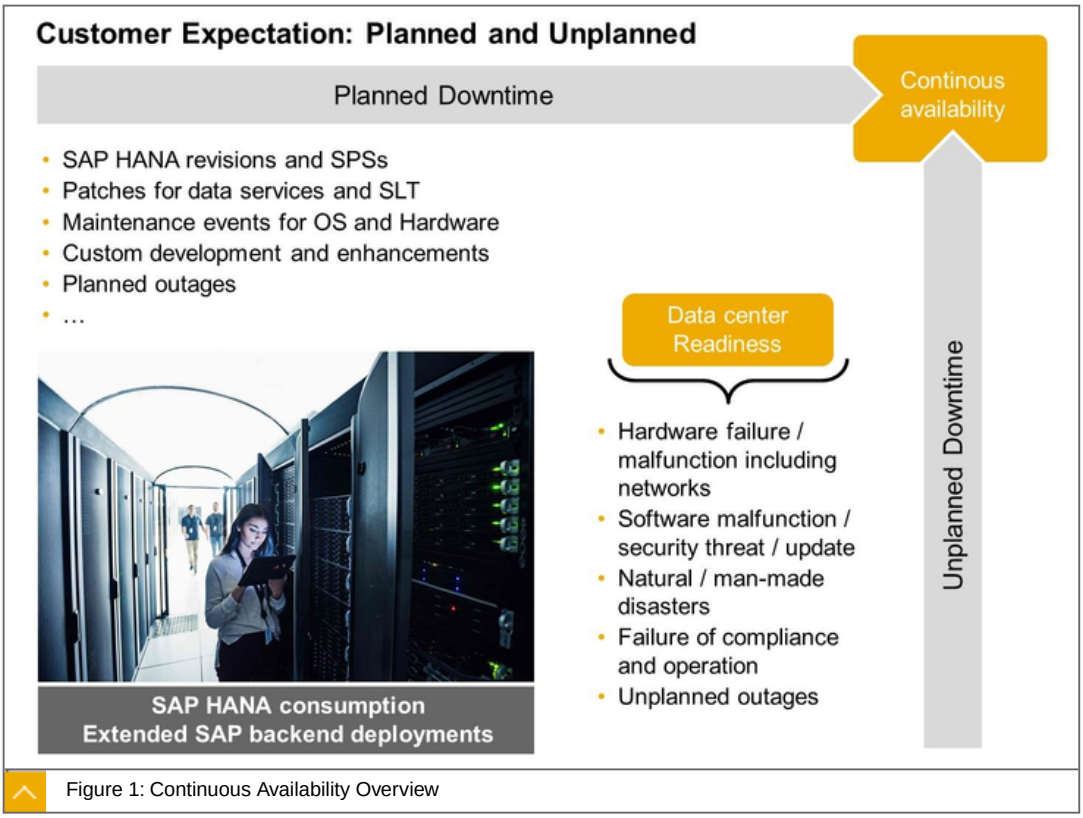


Figure 1: Continuous Availability Overview

The SAP HANA database platform is designed with high availability and disaster tolerance in mind. SAP HANA supports a broad range of recovery scenarios, from simple software errors or hardware failures, to disasters that take out an entire site.

What is High Availability?

Availability is usually indicated as a percentage of the operational uptime of a system, measured over the course of a year. For example, if a system is designed to be available 99.99% of the time (sometimes called "four nines"), its downtime per year must be less than 0.01%, or 52 minutes and 56 seconds.

That means less than an hour of downtime per year. This can be a very challenging target. To meet such challenging targets, high availability and disaster tolerance should be an integral part of the architectural design, that is, implemented on every layer of the infrastructure.

Downtime is the consequence of outages, which may be planned downtime (such as that for system upgrades or hardware replacements) or caused by unplanned downtime (such as that for software or hardware failures). Unplanned downtime can be triggered by equipment malfunction, software, or network failures, or a major disaster such as a fire, earthquake, a regional power loss, or a construction accident which may decommission the entire data center.

High Availability is a set of techniques, engineering practices, and design principles for business continuity. This is achieved by eliminating single points of failure (fault tolerance), and providing the ability to rapidly resume operations after a system outage with minimal business loss (fault resilience).

Fault Recovery is the process of recovering and resuming normal operations after an outage due to a fault.

Disaster Recovery is the process of recovering operations after an outage due to a prolonged data center or site failure. Preparing for disasters may require backing up data across longer distances, and may thus be more complex and costly.

Recovery - Key Performance Indicators (KPIs)

Customers commonly use two key measures to specify the recovery parameters of a system following an outage, the Recovery Point Objective (RPO) and the Recovery Time Objective (RTO). The RPO and RTO of a system are illustrated in the following figure.

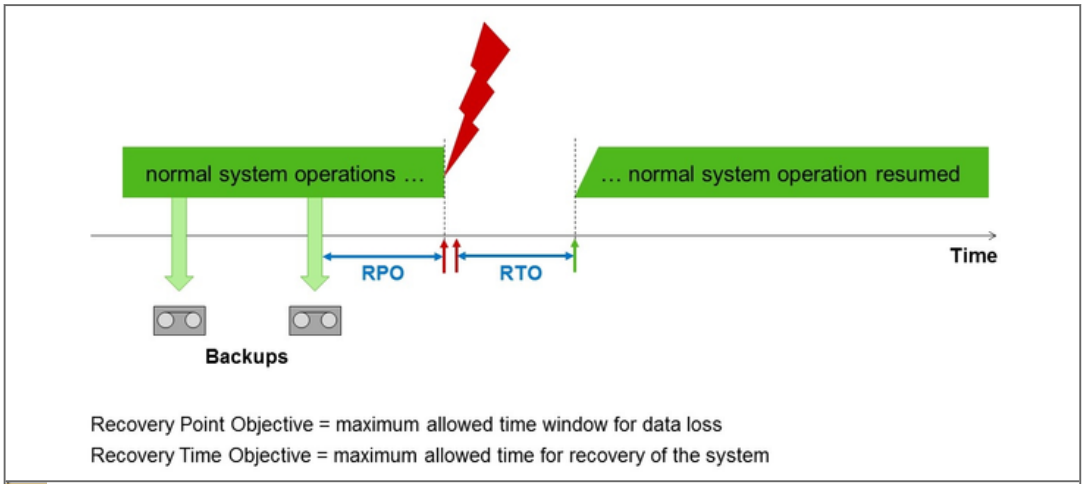


Figure 2: RPO and RTO

The RPO is the maximum permissible amount of time during which operational data may be lost without the ability to recover. It is the time between the last backup (data or log) and the crash. Almost every customer tries to achieve an RPO of 0, because the loss of business data is unacceptable.

The RTO is the maximum permissible amount of time it takes to recover the system, so that normal operations can resume. Many companies aim for a near-zero RTO, because during the RTO period, normal business is interrupted. Interrupted business leads to loss in revenue, which should be avoided as much as possible.

Eliminating Single Points of Failure

The key to achieving fault tolerance is to eliminate single points of failure by introducing redundancy. SAP HANA hardware vendors deliver several levels of redundancy to avoid outage due to component failure.

Generally speaking, these techniques are transparent to SAP HANA's operation. Nevertheless, they form a crucial line of defense against avoidable system outage, and therefore greatly contribute to business continuity.

Hardware Redundancy

SAP HANA hardware vendors design multiple layers of redundancy in their hardware components and subsystems. These include redundant and hot-swappable Power Supply Units (PSUs), fans, network interface cards, and enterprise-grade, error-correcting code memory.

These subsystems are designed in such a way that the redundant components can sustain operations of the system even when other components fail.

The storage system is particularly critical. Enterprise-grade storage systems combine multiple physical drives into logical units, with built-in standard Redundant Array of Independent Disks (RAID) techniques for redundancy and error recovery. These include mirroring (the writing of the same data to two different drives in parallel) and parity (the writing of extra bits to allow the detection and automatic correction of errors).

Network Redundancy

Redundant networks, network equipment, and network connectivity are required to avoid network failures affecting system availability. This is typically accomplished by deploying a completely redundant switch topology, using the Spanning Tree Protocol (STP) to avoid loops.

Routers can be configured with the Hot Standby Router Protocol (HSRP) for automatic failover. The Border Gateway Protocol (BGP) is commonly used to manage dual WAN connections.

Data Center Redundancy

Data centers that host SAP HANA solutions are equipped with Uninterrupted Power Supply (UPS) units and backup power generators, redundant cooling systems, and multi-sourced providers of network connectivity and electricity. This is done to achieve operational availability in the presence of individual failures, and the significant reduction of the probability of a business-impacting outage. Some enterprises operate fully duplicated data centers, providing a high level of disaster tolerance.

SAP HANA High Availability Support

As an in-memory database, SAP HANA must not only concern itself with maintaining the reliability of its data in the event of failures. It should also be concerned with resuming operations as quickly as possible with most of its data loaded back into memory.

The following figure shows the phases of SAP HANA High Availability support.

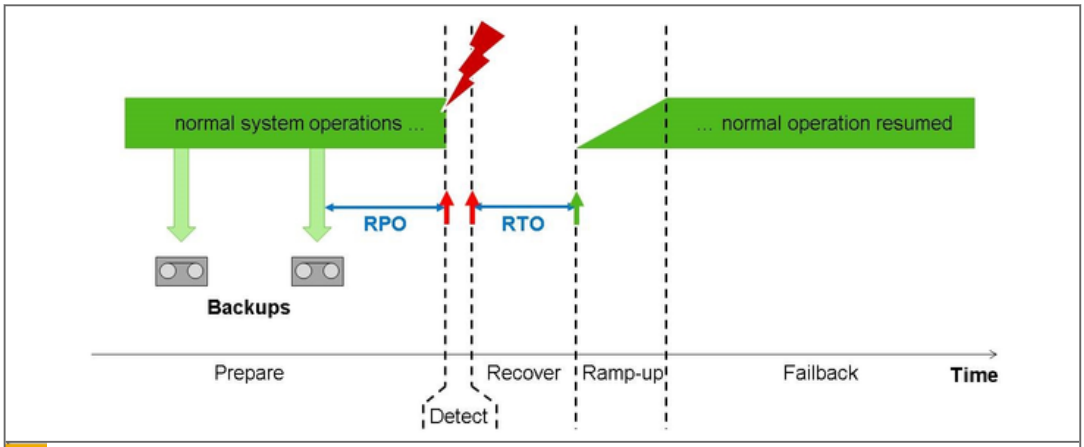


Figure 3: SAP HANA RPO and RTO Support

Prepare Phase

This first phase means being ready for disaster. During this time, the database is regularly backed up (data and log backups). The local or remote standby systems are operational and ready to take over.

This phase is often taken for granted, because everything is working within the defined parameters. Due to this relaxed attitude, some check procedures might be skipped. This is a disaster waiting to happen. Ensure that there are checks in place.

Detect Phase

Before a takeover can be initiated, a fault must be detected. This fault detection can be done automatically or manually. In both cases, false positives must be avoided, so a failure must be re-tested.

Try to keep the detect phase as short as possible to avoid the loss of revenue while systems are down.

Recover Phase

When a real failure has been detected, the takeover is triggered. Depending on the fault, different recovery processes can be triggered. The different recovery processes have different recovery runtimes. The runtime is also heavily dependent on the available hardware resources.

Ramp-up Phase

As soon as recovery is completed, the system is available in a ramp-up state. This is due to the fact that not all the data is loaded into memory yet and external interfaces might still be initializing.

To optimize this phase, you could investigate which data is needed most, so that this data can be loaded first.

Failback Phase

When all of the other phases are complete, the fault needs to be repaired. This can be a hardware repair or software updates. Both take time and may require additional testing before being applied to the production system.

When these repairs are done, the system may need to failback to the original data center and hardware. This can be triggered immediately or at the next data center maintenance window. If and when this failback is triggered is up to the customer, because this depends on contracts and service level agreements with third-party vendors and may involve additional costs.

SAP HANA Recovery Features

Different RPO and RTO values can be associated with different kinds of faults. Business-critical systems are expected to operate with an RPO of zero data loss in the case of local faults, and often even in the case of a disaster.

The challenges of disaster recovery are different for locally recoverable faults compared to total disasters. To achieve zero RPO and low RTO in a total disaster, data must be replicated synchronously over longer distances, which impacts regular system performance and may require more expensive standby and failover solutions.

All of this leads to trade-off decisions around the attributes of fault recovery functionality, cost, and complexity. SAP offers complementary design options, including three levels of disaster recovery support and three levels of automatic fault recovery support. These are summarized in the following table. More details on fault recovery and disaster recovery are provided in the next units of this course.

Recovery type	Recovery feature	Cost involved	RPO (data loss)	RTO (time)
Fault recovery support	Service Auto-Restart	No costs	0	Short
	SAP HANA Auto-Restart	No costs	0	Long
	Host Auto-Failover	Medium costs	0	Medium
Disaster recovery support	Backups	Low costs	0	Long
	Storage Replication	High costs	0	Medium
	System Replication	High costs	0	Short
	System Replication – Active/Active	Medium costs	0	Short
	System Replication – without data preload	Low costs	0	Medium

Figure 4: SAP HANA Recovery Features

Fault Recovery Support

Local faults, such as hardware and software failures, can often be handled in the same data center and hardware. Possible solutions to repair the error include restarting a failing service on the same server, or switching to a new host in the same data center. Such solutions can be implemented at almost no extra cost, as they are often a default part of the software and hardware solution provided by hardware vendors.

Service Auto-Restart

In the event of a software failure of one of the configured SAP HANA services (Index Server, Name Server, and so on), the failing service is restarted by the SAP HANA service auto-restart watchdog function.

This watchdog function is provided by the SAP HANA daemon process, which automatically detects the failure and restarts the stopped service process. Upon restart, the service loads data into memory and resumes its function. While all data remains safe, the service recovery takes some time.

SAP HANA Auto-Restart

The SAP HANA database system can be configured in an auto-restart mode. This can be useful after a power failure. When the power returns and the Linux operating system has been started successfully, the SAP HANA database system automatically performs a startup and recovery. The SAP HANA database system is available again for normal operations as soon as the startup and recovery is finished.

Host Auto-Failover

This is a local fault recovery solution that can be used in addition to, or as an alternative measure to, system replication. One or more hosts are added to an SAP HANA database system. These additional hosts are configured to work in standby mode.

As long as they are in standby mode, the databases on these hosts do not contain any data and do not accept requests or queries. This means these additional standby hosts cannot be used for other purposes, such as quality or test systems.

Disaster Recovery Support

Backups

SAP HANA is an in-memory database, but all the data is persisted on disk as well. Data is persisted on disk by means of regular savepoints. These savepoints are performed by default every five minutes. In between these savepoints, all the changes are recorded in transaction redo logs.

To make sure that SAP HANA can recover from hardware failures, regular data and log backups must be performed. These data and log backups must be shipped to the secondary site to make sure that the system can recover from a total disaster.

Storage Replication

This is a method to continuously replicate all persisted data and log information to the secondary site. Several SAP HANA hardware partners offer a storage-level replication solution, which delivers a backup of the volumes or file-system to a remote, networked storage system.

System Replication

This is a native SAP HANA high availability solution that provides a continuously-replicated SAP HANA system on the secondary site. The data is already loaded into memory, so takeover times are short in comparison to backup and storage replication solutions.

System Replication Active/Active

This is a second native SAP HANA system replication solution that allows the data to be read from the secondary system. In this setup, the secondary system can be used to handle the reporting workload without disrupting the primary system.

System Replication without Data Preload

This is a third native SAP HANA scenario. In this solution, the secondary system does not pre-load data, and hence consumes very little memory. This allows the hosts of the secondary system to serve dual purposes. For example, for development, unit testing, or QA with separate storage. Before takeover, these activities must of course be turned off. The trade-off in this scenario is a longer RTO in the case of failover.

LESSON SUMMARY
You should now be able to:

Understand the different SAP HANA high availability features

Unit 1
Lesson 2

Exploring Disaster Recovery in SAP HANA

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Describe the disaster recovery features in SAP HANA

Disaster Recovery Support

Business Example

As a SAP HANA database administrator, you are responsible for your company's SAP ERP and SAP Business Warehouse (SAP BW) systems. You need to understand which disaster recovery features are supported by the SAP HANA database.

Backups

Backups are one of the key disaster recovery features offered by SAP HANA.

SAP HANA uses in-memory technology, but of course, it fully persists any transaction that changes the data, such as row insertions, deletions, and updates, so it can resume from a power outage without loss of data. SAP HANA persists two types of data to the storage system, transaction redo logs and data changes in the form of savepoints.

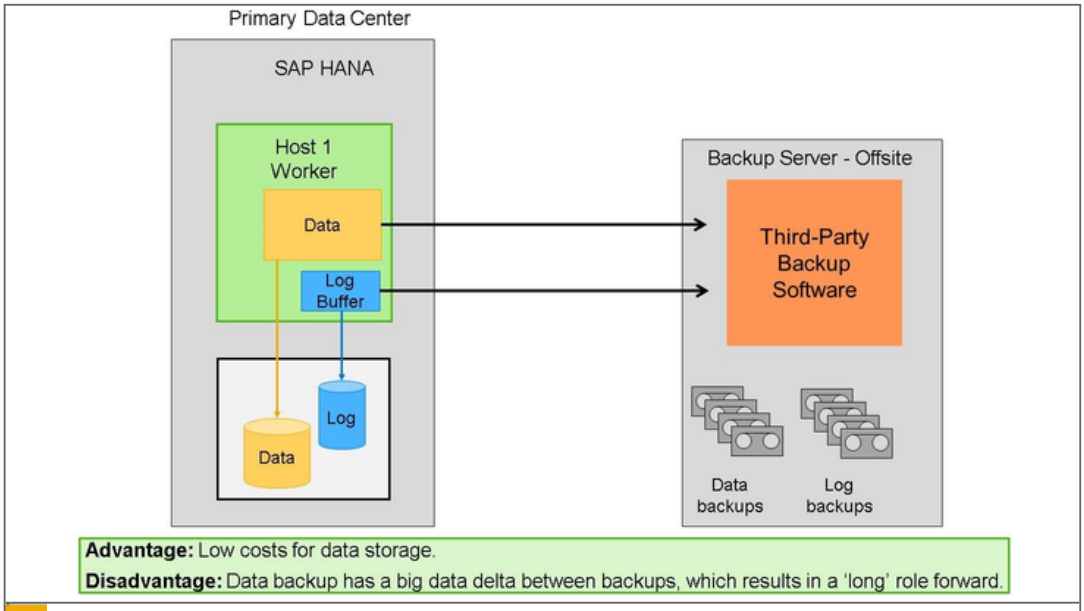


Figure 5: SAP HANA Backups

A transaction redo log is used to record a change. To make a transaction durable, it is not required to persist the complete data when the transaction is committed. Instead, it is sufficient to persist the redo log. On failure, the most recent consistent state of the database

can be restored by replaying the changes recorded in the log, redoing completed transactions, and rolling back incomplete transactions.

A savepoint is a periodic point in time when all the changed data is written to storage, in the form of pages. One goal of performing savepoints is to speed up restart. When starting a system, logs need not be processed from the beginning, but only from the last savepoint position. Savepoints are coordinated across all processes (called SAP HANA services) and instances of the database to ensure transaction consistency. By default, savepoints are performed every five minutes, but this value is configurable.

Savepoints normally overwrite older savepoints, but it is possible to freeze a savepoint for future use. This is called a snapshot. Snapshots can be replicated in the form of full data backups, which can be used to restore a database to a specific point in time. This can be useful in the event of data corruption, for instance. In addition to data backups, smaller periodic log backups ensure the ability to recover from fatal storage faults with minimal loss of data.

Savepoints can be saved to local storage, and the additional backups can be saved to backup storage. Local recovery from the crash uses the latest savepoint, and then replays the last logs, to recover the database without any data loss. If the local storage was corrupted by the crash, it is still possible to recover the database from the data and log backups, possibly with some loss of data. Regularly shipping backups to a remote location over a network or using couriers can be a simple and relatively inexpensive way to prepare for a disaster. Depending on the frequency and shipping method, this approach may have a recovery time ranging from hours to days.

Storage Replication

SAP HANA offers disaster recovery support for storage replication solutions provided by hardware partners.

One drawback of backups is the potential loss of data between the time of the last backup and the time of the failure. A preferred solution is to provide continuous replication of all persisted data. Several SAP HANA hardware partners offer a storage-level replication solution that delivers a backup of the volumes or file system to a remote, networked storage system. In some of these vendor-specific solutions, which are certified by SAP, the SAP HANA transaction only completes when the locally persisted transaction log has been replicated remotely. This is called synchronous storage replication. Synchronous storage replication can be used only where the distance between the primary and backup site is relatively short (typically 100 kilometers or less), allowing for sub-millisecond round-trip latencies.

Due to its continuous nature, storage replication (sometimes also called remote storage mirroring) can be a more attractive option than backups, as it reduces the amount of time between the last backup and a failure. Another advantage of storage replication is that it also enables a much shorter recovery time. This solution requires a reliable, high-bandwidth and low-latency connection between the primary site and the secondary site.

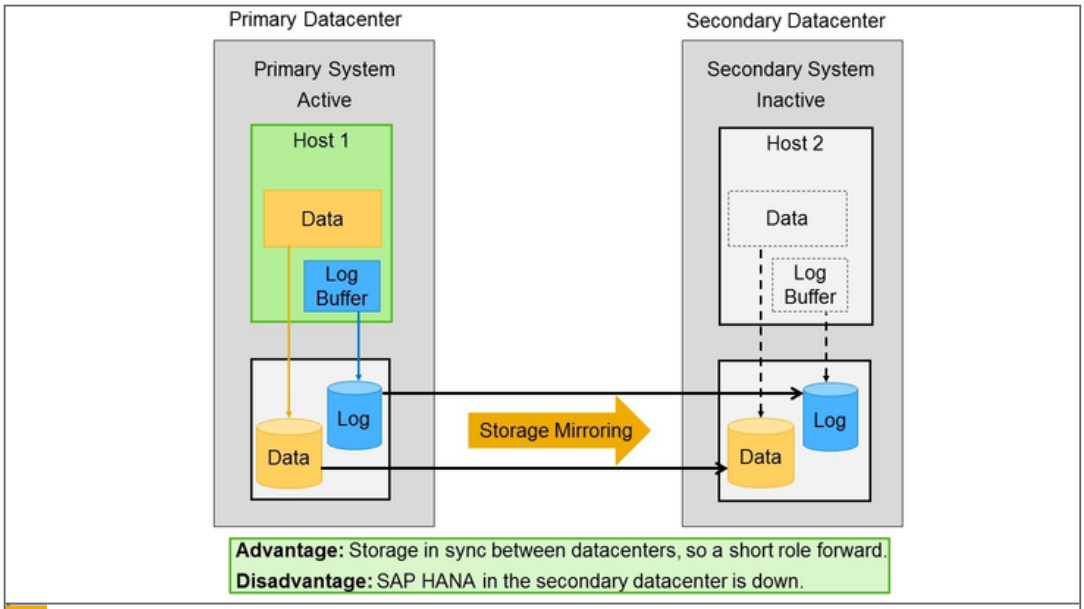


Figure 6: SAP HANA Storage Replication

Due to its continuous nature, storage replication (sometimes called “remote storage mirroring”) offers a more attractive RPO than backups, but this solution of course requires a reliable, high bandwidth, and low latency connection between the primary site and the secondary site.

In the event of a total disaster that justifies full system failover, an administrator attaches a standby system to the replicated storage. The administrator then restarts the SAP HANA system. The administrator must ensure that the failed primary system can no longer write to the replicated storage by fencing it from the primary system. If the primary system is not fenced, then there is a risk of data corruption. This corruption can occur because two systems can write to the same storage at the same time.

System Replication

System replication is available in every SAP HANA installation offering inherent disaster recovery support.

System replication is an alternative high availability solution for SAP HANA that provides an extremely short RTO, and is compatible with all SAP HANA hardware partner solutions. System replication employs an "N+N" approach, with a secondary standby SAP HANA system with the same number of active nodes as the active, primary system. Each service and instance of the primary SAP HANA system communicates pairwise with a counterpart in the secondary system.

The secondary system can be located near the primary system to serve as a rapid failover solution for planned downtime, or to handle storage corruption or other local faults. Alternatively, it can be installed in a remote site to be used in a disaster recovery scenario.

In addition, both approaches can be chained together with multi-tier system replication. Like storage replication, this disaster recovery option requires a reliable connection channel between the primary and secondary sites. The instances in the secondary system operate in recovery mode. In this mode, all secondary system services constantly communicate with their primary counterparts, replicate and persist data and logs, and load data to memory. The main difference to primary systems is that the secondary systems do not accept requests or queries.

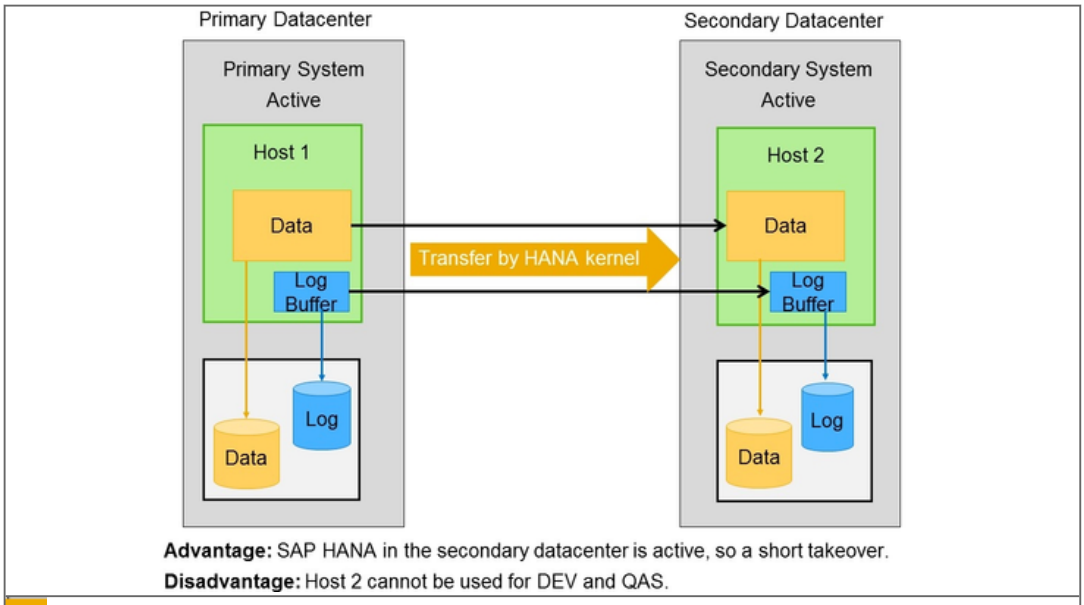


Figure 7: SAP HANA System Replication

When the secondary system is started in recovery mode, each service component establishes a connection with its counterpart, and requests a snapshot of the data in the primary system. From then on, all logged changes in the primary system are replicated. Whenever logs are persisted in the primary system, they are also sent to the secondary system. A transaction in the primary system is not committed until the logs are replicated.

Explaining the HA201 System Landscape

The system landscape created for the HA201 course is a landscape with five Linux servers. Four of these servers are used for installations of the high availability SAP HANA scale-out system and the SAP HANA system replication systems. The fifth server is an SAP HANA cockpit 2.0 system that is used to configure and monitor the installed high availability and disaster recovery systems.

During the exercises, we simulate crashes of SAP HANA nodes, so that we can monitor what happens to the rest of the SAP HANA system. Every group of four participants gets a landscape as shown in the following figure.

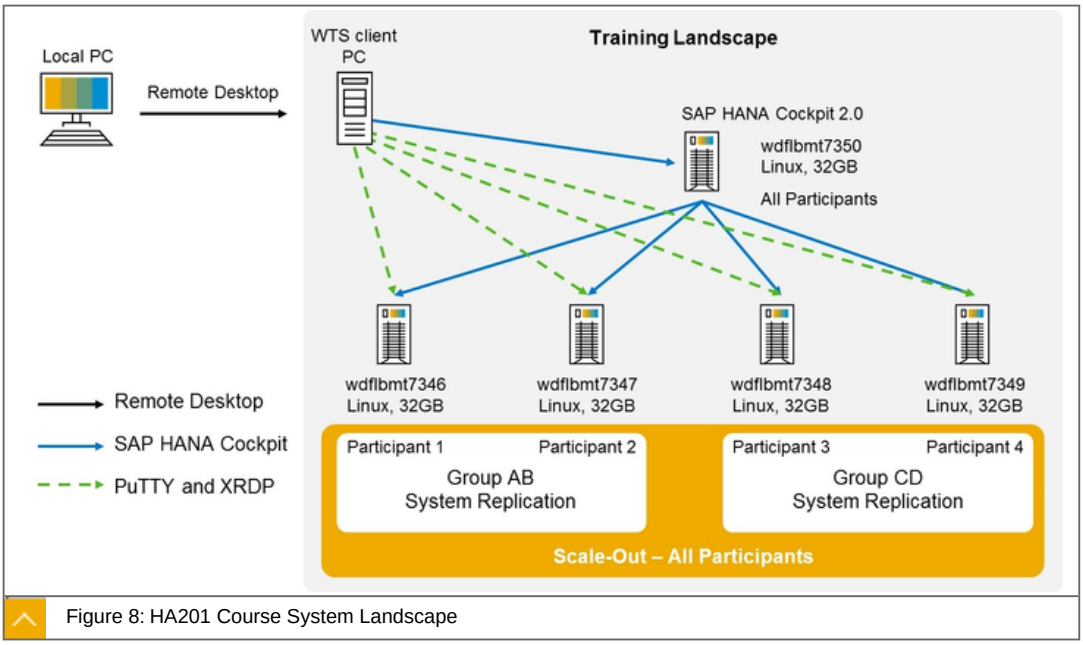


Figure 8: HA201 Course System Landscape

During the scale-out and tenant administration exercises, all the Linux servers are grouped together into one SAP HANA scale-out system with four nodes. This four-node scale-out system is installed and re-configured, and failures of slave and master nodes are simulated.

In this setup, every participant has a Linux server to work through the exercises. As some steps of the exercises can only be performed one at a time, or even only once, these steps are evenly divided over the groups of four participants.

During the system replication exercises, the four Linux servers are split into two sets of two servers. Participants 1 and 2 become Group AB, and participants 3 and 4 become Group CD. In every group, the SAP HANA system replication setup is installed, configured and tested. In the last exercise, the SAP HANA system replication with Active/Active is set up and tested.

LESSON SUMMARY
You should now be able to:

- Describe the disaster recovery features in SAP HANA

Unit 1
Lesson 3

Exploring Fault Recovery in SAP HANA

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Explain the fault recovery features of SAP HANA

Fault Recovery in SAP HANA

Business Example

As an SAP HANA database administrator, you are responsible for your company's SAP ERP and SAP Business Warehouse (BW) systems. You need to understand the SAP HANA startup framework, and how the SAP HANA database handles hardware and software faults.

Service Auto-restart

In the event of a software failure that disables one of the configured SAP HANA services (index server, name server, and so on), the failing service is restarted by the SAP HANA Service Auto-Restart watchdog function, which automatically detects the failure and restarts the stopped service process. Upon restart, the service loads data into memory and resumes its function. While all data remains safe (RPO=0), the service recovery takes some time.

The restarting of the failing SAP HANA services is handled by the SAP HANA daemon service. One of the tasks of this service is to watch over the other HDB services and restart them if necessary. The SAP HANA daemon itself is started by the SAP HANA startup framework. This framework resembles the SAP NetWeaver startup framework and is shown in the following figure.

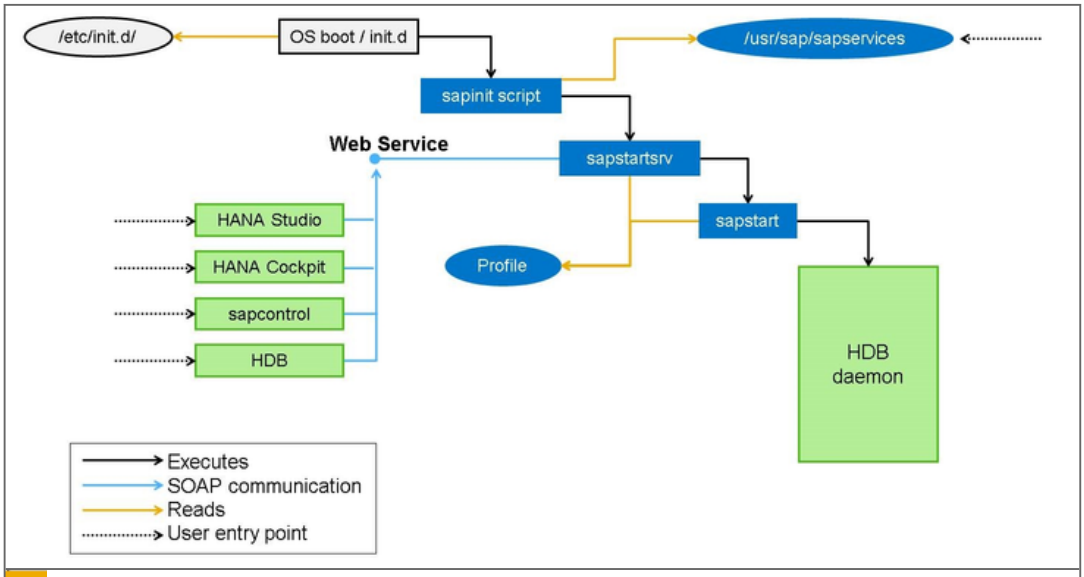


Figure 9: SAP HANA Startup Framework

When a Linux server boots, the boot process goes through several stages that are executed by different components:

1. **BIOS/UEFI**

   After turning on the computer, the BIOS or the UEFI initializes the different basic hardware components, such as the screen and keyboard, and tests the main memory. As soon as the first bootable hard disk is identified, the BIOS/UEFI passes control to the boot loader.

2. **Boot Loader**

   Located on the first data sector of the first hard disk, the master boot record is loaded into the main memory. On Linux systems, this boot loader is usually GRUB 2. When the boot loader is finished, it passes control to the operating system.

3. **Operating System**

   When the boot loader passes control to the operating system, the Linux kernel and initial RAM-based file system (initramfs) are loaded into memory.

4. **init process**

   From the initramfs, the init executable is started, and then it mounts the root file system. When the root file system is successfully mounted, control is passed to the systemd daemon. The initramfs file system is cleared.

5. **systemd daemon**

   The systemd daemon takes care of the booting of the rest of the operating system. The systemd daemon mounts all the defined file systems and starts the required services. When this is finished, the Linux operating system is available for the user.

   When a Linux server boots, as described in the previous steps, the systemd daemon identifies into which “target” (formerly known in System V as “runlevel”) the server needs to be started. When the target runlevel is identified, the systemd daemon starts the required programs belonging to that target runlevel.

   An overview of the available target runlevels on the server can be generated with the command: **systemctl list-units --type=target** .

   In the different target runlevels, only the required start scripts, programs, or daemons are started. One of these start scripts is the sapinit script found in the /etc/init.d folder. The sapinit script is installed on the server during the SAP HANA installation.

1. The **sapinit** script reads the /usr/sap/sapservices file and starts the sapstartsrv daemon.
2. The **sapstartsrv** then reads the SAP HANA instance profile and starts the sapstart executable.
3. The **sapstart** program reads the SAP HANA instance profile also to see if the SAP HANA database needs to be started automatically.

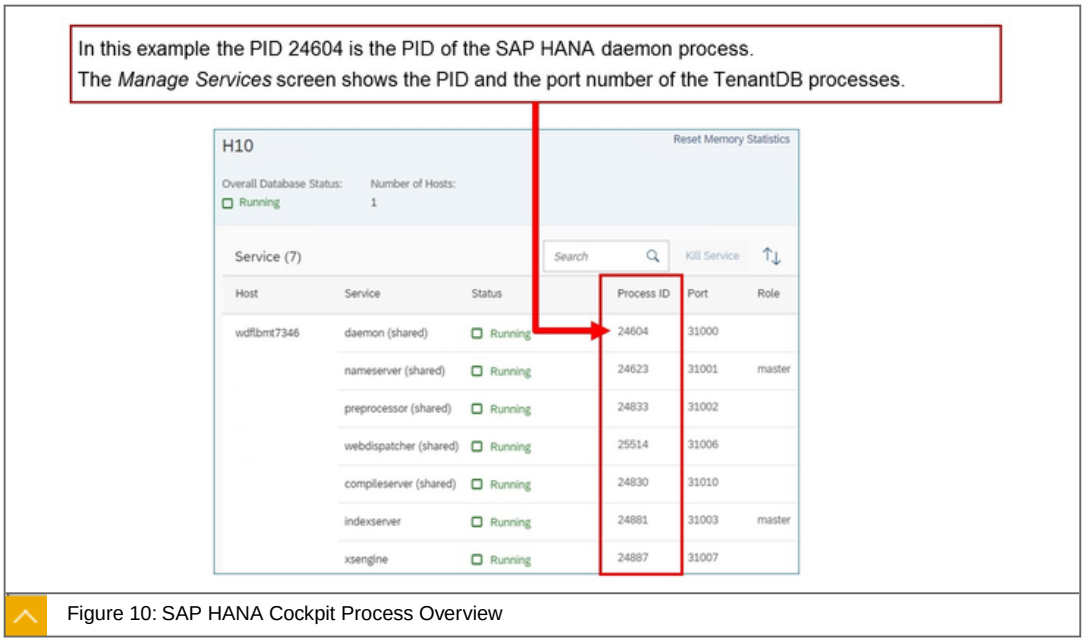


Figure 10: SAP HANA Cockpit Process Overview

From the SAP HANA cockpit 2.0 in the Manage Services application, all the running SAP HANA services are shown. The column Process ID is very interesting here, as it shows the process ID (PID) of the SAP HANA services. The PID of the daemon process is also shown. This PID can also be found at the Linux operating system level. In this way, you can easily identify the SAP HANA services at the operating system level.

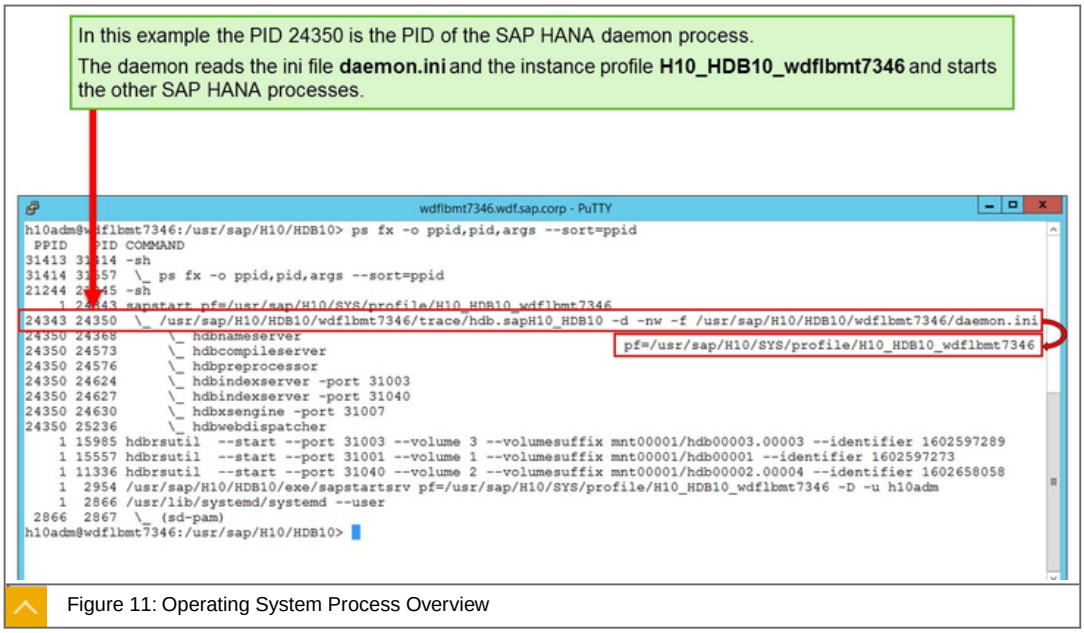


Figure 11: Operating System Process Overview

The process list can also be shown at the Linux operating system level. Showing processes under Linux can be done in many ways, for example, using the command ps fx -o ppid,pid,args --sort=ppid .

This command not only shows the PIDs of all the SAP HANA services, but also their starting order and hierarchy. In the previous figure, it is clearly visible that the init process starts sapstart.

The sapstart process in turn starts the SAP HANA daemon process. In the operating system process overview, this process is not called daemon, but by looking at the PID you can see that it is indeed the SAP HANA daemon.

The SAP HANA daemon, often referred to in the documentation as hdbdaemon, is responsible for starting all the other SAP HANA services such as:

- hdbnameserver
- hdbcompileserver
- hdbpreprocessor
- hdbindexserver
- hdbxsengine
- hdbwebdispatcher

This previous list is not fixed, as it depends on the SAP HANA version and on the SAP HANA tenant configuration.

SAP HANA Autostart

The SAP HANA database can be started, stopped, and restarted at the Linux operating system level. This is often needed to automate tasks. The scripts need the information contained in the startup profile.

The startup profile can be found in the location /usr/sap/<SID>/SYS/profile . The startup profile lists the SAPSYSTEMNAME, SAPSYSTEM, INSTANCE_NAME, and SAPLOCALHOST, but none of these parameters should be modified.

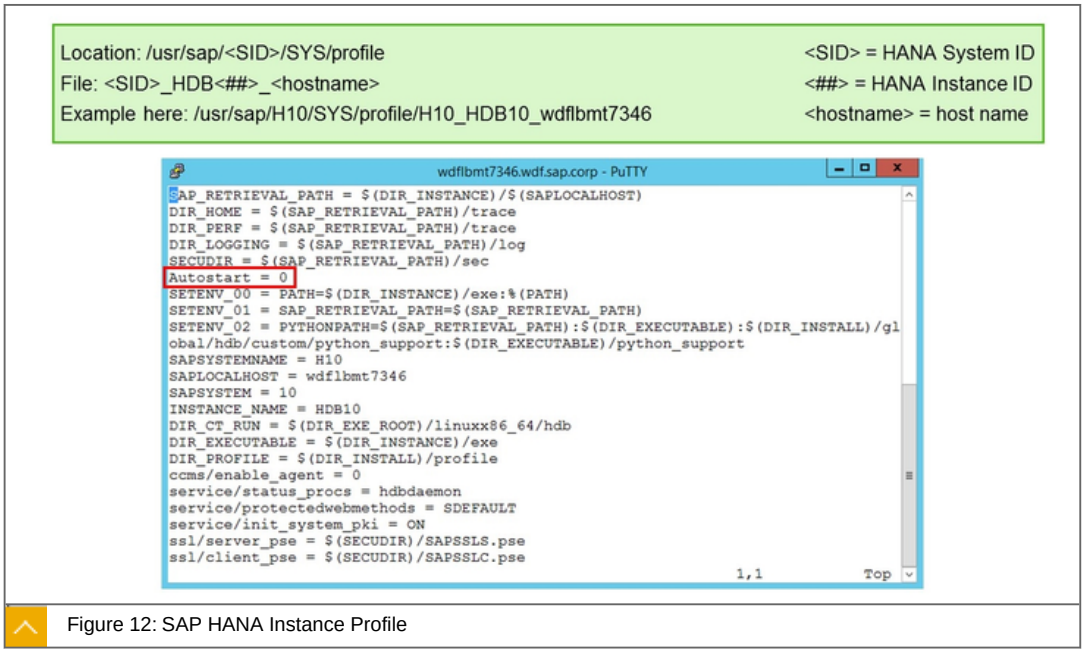


Figure 12: SAP HANA Instance Profile

The only exception is the Autostart parameter. This parameter controls the automatic start of the SAP HANA database by the sapstart process.

If Autostart=0, the SAP HANA database does not automatically start when the operating system starts.

This can be very useful if there are several SAP HANA databases (such as, end-user training and test) installed on one server that should be manually stopped and started as needed by the system administrator.

If Autostart=1, the SAP HANA database starts automatically when the operating system starts.

This can be very useful for a production system that needs to be available as soon as possible after a hardware failure, or for an SAP HANA database that should be automatically available after the scripted deployment of a fresh virtual image.

Host Auto-failover

Host auto-failover is a local "N+m" (m is often 1) fault recovery solution that can be used as a supplemental or alternative measure to the system replication solution described earlier. One (or more) standby hosts are added to an SAP HANA system and configured to work in standby mode. As long as they are in standby mode, the databases on these hosts do not contain any data and do not accept requests or queries.

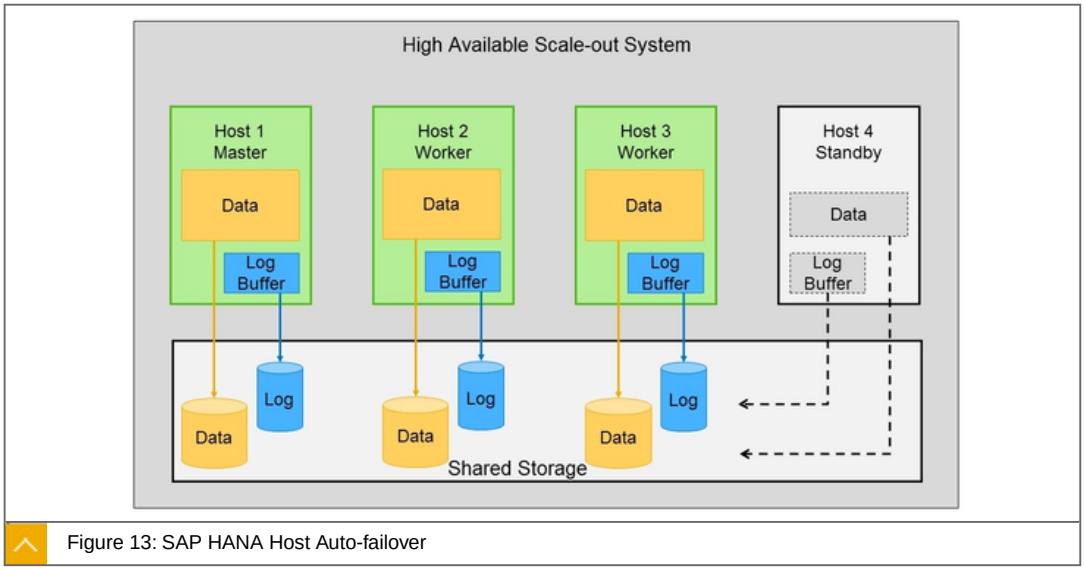


Figure 13: SAP HANA Host Auto-failover

When an active (worker) host fails, a standby host automatically takes its place. Because the standby host may take over operation from any of the primary hosts, it needs access to all the database volumes. This can be accomplished by using a shared, networked storage server, by using a distributed file system, or with vendor-specific solutions that use an SAP HANA programmatic interface (the Storage Connector API) to dynamically detach and attach (mount) networked storage (for example, using block storage over Fiber Channel) upon failover.

LESSON SUMMARY
You should now be able to:

Explain the fault recovery features of SAP HANA

Learning Assessment

1. Which of the following are the two Key Performance Indicators (KPIs) for a system recovery after a failure?

   Choose the correct answers.

   - ☐ **A** PSU
   - ☐ **B** BGP
   - ☐ **C** RPO
   - ☐ **D** RTO

2. What are SAP HANA High Availability support phases?

   Choose the correct answers.

   - ☐ **A** Prepare phase
   - ☐ **B** Disaster phase
   - ☐ **C** Downtime phase
   - ☐ **D** Recover phase

3. A savepoint is executed at every commit to make sure that every change is persisted on disk.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

4. What are the native SAP HANA disaster recovery features?

   Choose the correct answers.

   - ☐ **A** Backups
   - ☐ **B** System Replication
   - ☐ **C** Storage Replication
   - ☐ **D** Host Auto-Failover

5. Only backups are offered in the SAP HANA database for disaster recovery.

   Determine whether this statement is true or false.

   - [ ] True
   - [ ] False

6. In the event of a software failure, sapstartsrv restarts the failed SAP HANA services.

   Determine whether this statement is true or false.

   - [ ] True
   - [ ] False

7. Which services are part of the SAP HANA Startup Framework?

   Choose the correct answers.

   - [ ] **A** sapcontrol
   - [ ] **B** sapstartsrv
   - [ ] **C** sapservices
   - [ ] **D** sapstart

8. Which services are part of the SAP HANA Startup Framework?

   Choose the correct answers.

   - [ ] **A** sapcontrol
   - [ ] **B** sapstartsrv
   - [ ] **C** sapservices
   - [ ] **D** sapstart

Learning Assessment - Answers

1. Which of the following are the two Key Performance Indicators (KPIs) for a system recovery after a failure?

   Choose the correct answers.

   - [ ] **A** PSU
   - [ ] **B** BGP
   - [X] **C** RPO
   - [X] **D** RTO

   You are correct! The two Key Performance Indicators (KPIs) for a system recovery after a failure are the Recovery Period Objective (RPO) and the Recovery Time Objective (RTO).

2. What are SAP HANA High Availability support phases?

   Choose the correct answers.

   - [X] **A** Prepare phase
   - [ ] **B** Disaster phase
   - [ ] **C** Downtime phase
   - [X] **D** Recover phase

   You are correct! The SAP HANA High Availability support phases are: the Prepare phase and the Recover phase.

3. A savepoint is executed at every commit to make sure that every change is persisted on disk.

   Determine whether this statement is true or false.

   - [ ] True
   - [X] False

   You are correct! A write to the transaction log is executed with every commit. A savepoint is executed, by default, every five minutes.

4. What are the native SAP HANA disaster recovery features?

   Choose the correct answers.

   [X] **A** Backups

   [X] **B** System Replication

   [ ] **C** Storage Replication

   [ ] **D** Host Auto-Failover

   You are correct! Backups and System Replication are the native SAP HANA disaster recovery features.

5. Only backups are offered in the SAP HANA database for disaster recovery.

   Determine whether this statement is true or false.

   [ ] True

   [X] False

   You are correct! Storage replication and system replication are also available for disaster recovery.

6. In the event of a software failure, sapstartsrv restarts the failed SAP HANA services.

   Determine whether this statement is true or false.

   [ ] True

   [X] False

   You are correct! In the event of a software failure, the HDBdaemon restarts the failed SAP HANA services.

7. Which services are part of the SAP HANA Startup Framework?

   Choose the correct answers.

   [ ] **A** sapcontrol

   [X] **B** sapstartsrv

   [ ] **C** sapservices

   [X] **D** sapstart

   You are correct! The services sapstart and sapstartsrv are part of the SAP HANA Startup Framework.

8. Which services are part of the SAP HANA Startup Framework?

   Choose the correct answers.

   - [ ] **A** sapcontrol
   - [X] **B** sapstartsrv
   - [ ] **C** sapservices
   - [X] **D** sapstart

   You are correct! The services sapstart and sapstartsrv are part of the SAP HANA Startup Framework.

UNIT 2

SAP HANA Fault Tolerance

UNIT OBJECTIVES

- Install a high availability SAP HANA system
- Introducing SAP HANA scale-out systems

Perform table partitioning tasks

Perform Table Placement Tasks

Reconfigure a scale-out SAP HANA system

Understand what happens during a failure of a slave node

Understand what happens during a failure of the master node

Remove a host from a scale-out system

Add a host to a scale-out system

Unit 2
Lesson 1

Installing High Availability SAP HANA

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Install a high availability SAP HANA system

Installation of a High Availability SAP HANA System

Business Example

As an SAP HANA database administrator, you need to install and administer your company's high availability scale-out SAP HANA systems. You need to have hands-on experience with installing SAP HANA single-host systems and extending such a system with additional hosts.

Installing a Multiple-Host System

The SAP HANA database lifecycle manager can be used to install an SAP HANA multiple-host system in one of the program interfaces, and with a combination of parameter specification methods.

A multiple-host system is a system with more than one host, which can be configured as active worker hosts or idle standby hosts. The SAP HANA software is built for a flexible architecture that allows a distributed installation. This means that the load can be balanced between different hosts. The SAP HANA software can be installed on a shared file system environment. This shared file system must be mounted by all hosts that are part of the system.

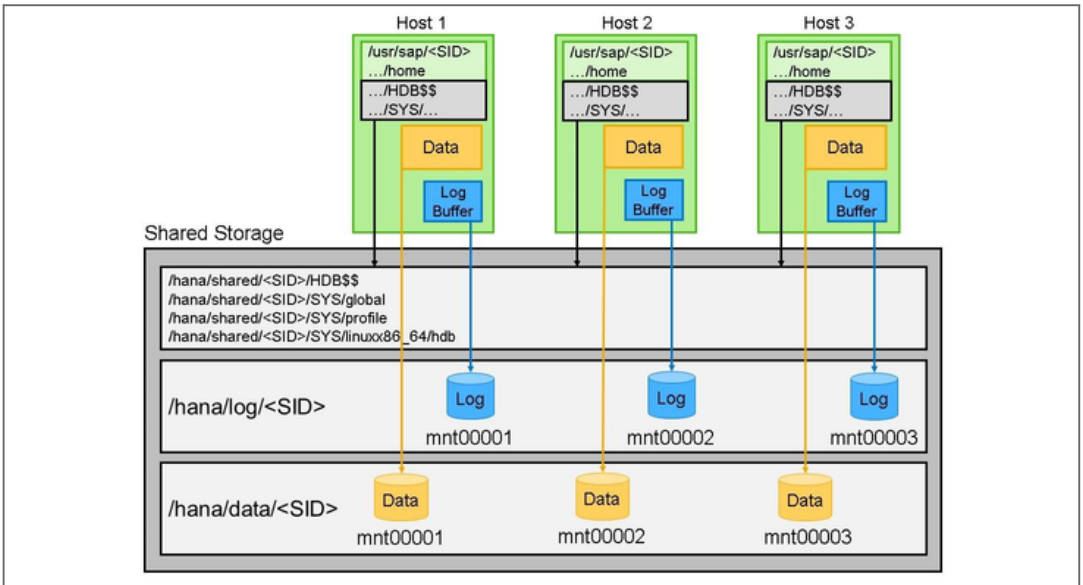


Figure 14: SAP HANA Multiple-Host System

A multiple-host system can be installed as a new system on several hosts, or by adding one or more new hosts to an already installed single-host system. To add hosts to an existing system, use the SAP HANA resident HDBLCM.

Multiple-Host System Concepts

It is important to review multiple-host system concepts, such as host grouping and storage options, before installing a multiple-host system.

Host Types

When configuring a multiple-host system, the additional hosts must be defined as **worker** hosts or **standby** hosts (worker is the default). Worker machines process data; standby machines do not handle any processing and instead just wait to take over processes in the case of worker machine failure.

Auto-Failover for High Availability

As an in-memory database, SAP HANA is not only concerned with maintaining the reliability of its data in the event of failures, but also with resuming operations with most of that data loaded back in memory as quickly as possible. Host auto-failover is a local fault recovery solution that can be used as a supplemental or alternative measure to system replication. One (or more) standby hosts are added to an SAP HANA system, and configured to work in standby mode.

Before installing a multiple-host system, it is important to consider whether high availability is necessary, and how hosts should be grouped to ensure preferred host auto-failover. For host auto-failover to be successful, if the active (worker) host fails, the standby host takes over its role by starting its database instance using the persisted data and log files of the failed host. The name server of one of the SAP HANA instances acts as the cluster manager that pings all hosts regularly. If a failing host is detected, the cluster manager ensures that the standby host takes over the role and the failing host is no longer allowed write access to the files (called “fencing”) to avoid data corruption. The crash of a single service does not trigger failover, because services are normally restarted by hdbdaemon.

High Availability Host Grouping

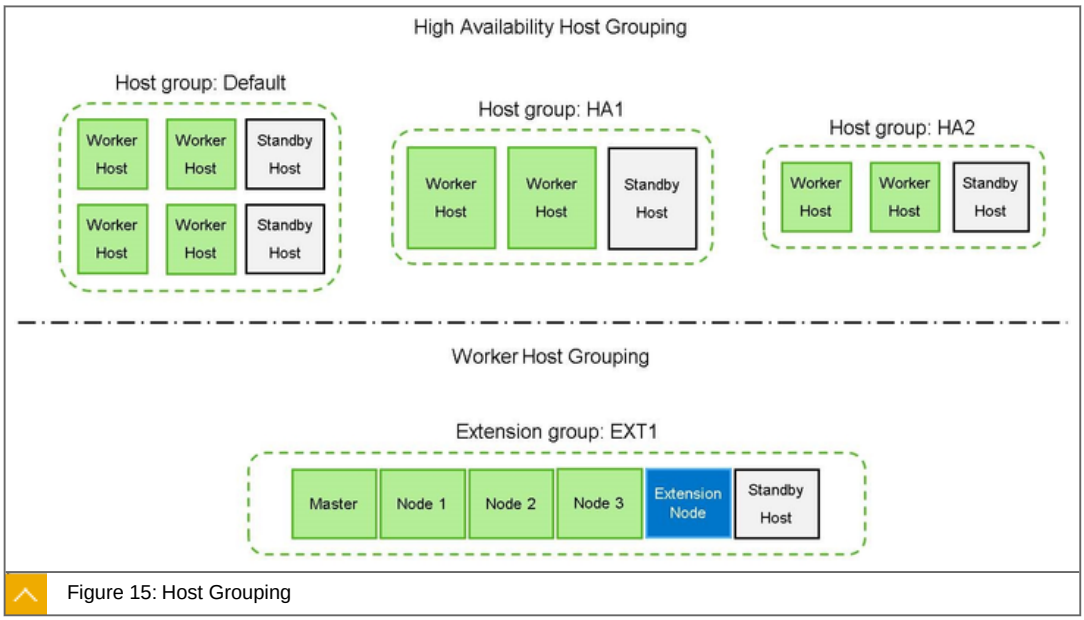


Figure 15: Host Grouping

Host grouping does not affect the load distribution among worker hosts; the load is distributed among all workers in an SAP HANA system. If there are multiple standby hosts in a system, host grouping should be considered, because host grouping decides the allocation of standby resources if a worker machine fails. If no host group is specified, all hosts belong to one host group called "default". The more standby hosts in one host group, the greater the failover security.

If each of the standby hosts is in a different host group, the standby host in the same group as the failing worker host is preferred. If a standby host is not available in the same host group, the system tries to fail over to a standby host that is part of another host group. The advantage of this configuration is that in an SAP HANA system with mixed machine resources, similar sized machines can be grouped together. If a small worker host fails, and a small standby in the same group takes over, the processes are moved to a machine with similar resources, which allows processing to continue as usual with optimal resource allocation.

Worker Host Grouping

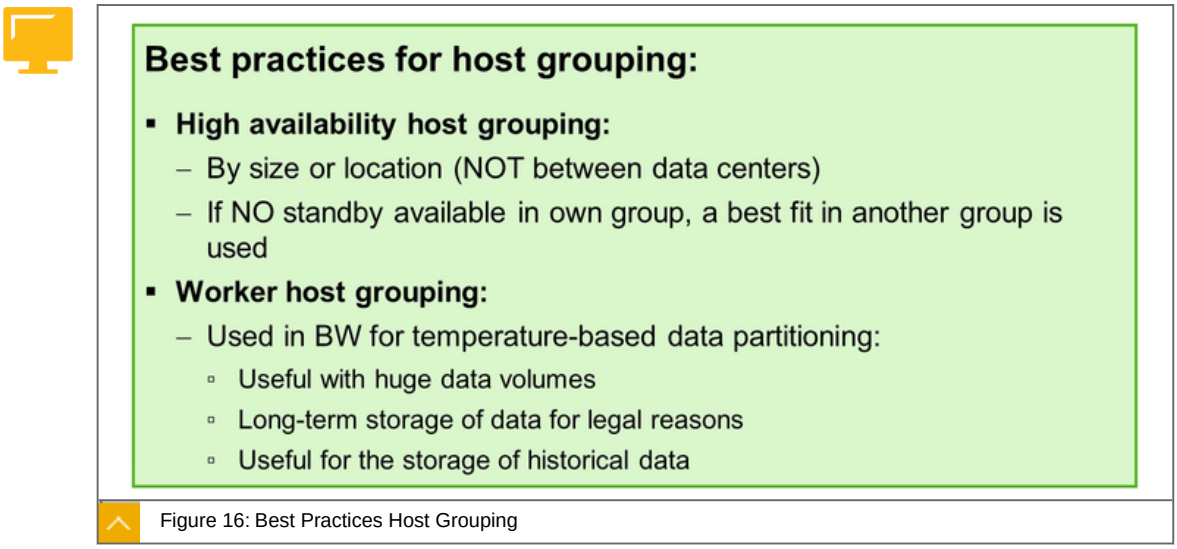


Figure 16: Best Practices Host Grouping

If you use SAP Business Warehouse to apply a temperature-based data strategy, you can significantly optimize the usage of memory and hardware resources by reserving one node of the scaled-out SAP HANA landscape exclusively for warm data. In information lifecycle management, multi-temperature strategies are often applied whereby data is classified by access frequency as either hot, warm, or cold. Depending on this classification and data usage, this data is stored in different memory areas.

A multi-temperature memory strategy may be required for different reasons, for example:

Storage of historical data.

Clickstream logs for multiple years of Web data and detailed machine logs.

Guidelines for saving company data, such as the need to save all data for at least seven years for legal reasons.

The standard SAP HANA sizing guidelines allow for a data footprint of 50% of the available RAM. This ensures that all data can always be kept in RAM, and there is sufficient space for intermediate result sets. These sizing guidelines can be significantly relaxed on the extension group, as warm data is accessed less frequently, with reduced performance SLAs, with less CPU-intensive processes, only partially at the same time.

To implement a multi-temperature memory strategy, you can assign hosts to worker groups. Hot and warm data are then distributed across hosts. To increase performance and memory usage, a worker node is assigned to a separate extension node. Unlike the standard nodes (master and worker), the extension node is intended exclusively for data that is not accessed as frequently (warm) as other data (hot). For more information, see SAP Note: 2453736.

Storage and File System Options

In single-host SAP HANA systems, it is possible to use local file systems residing on directly-attached internal or external storage devices, such as SCSI hard drives, SSDs, SAN storage, or NAS. However, to build a multiple-host system with failover capabilities, this is not sufficient. Either the chosen file system type or the SAN infrastructure, along with an SAP HANA functionality capable of disc fencing, must ensure the following:

- The standby host has file access to the data and log volumes of the failed host.
- The failed worker host no longer has access to write to files, called “fencing”.

There are two fundamentally different storage configurations that meet these two conditions: shared storage devices or separate storage devices with failover reassignment. Do not confuse shared storage with the installation directory /hana/shared that must be shared across all hosts.

Shared File Systems

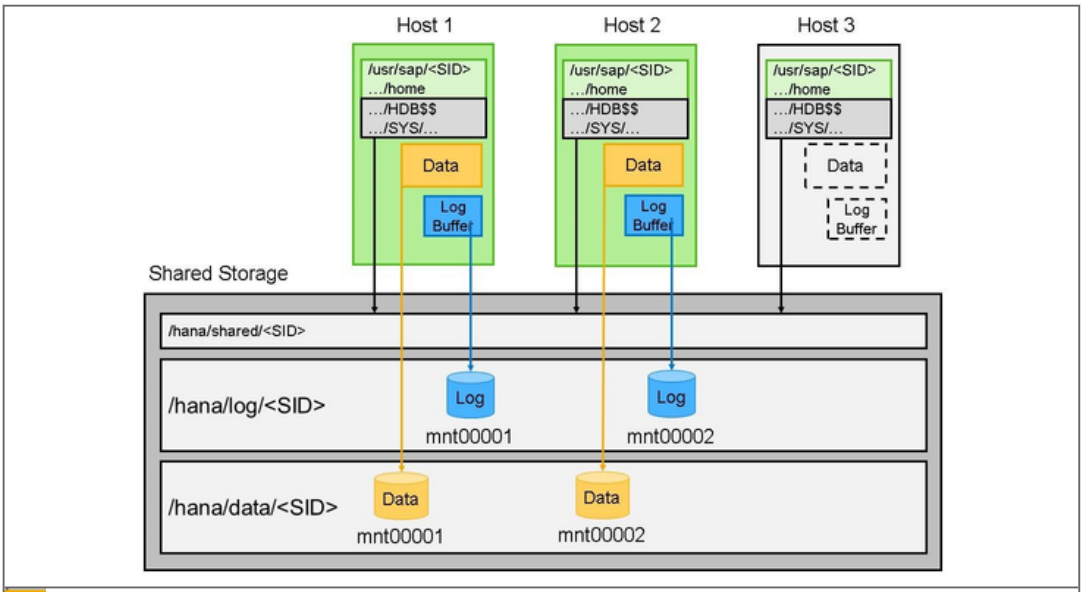


Figure 17: High Availability Using Shared Storage

A shared storage subsystem, which is accessed using file systems such as NFS or IBM's GPFS, makes it easy to ensure that the standby host has access to all active host files in the system. In a shared storage solution, the externally attached storage subsystem devices can provide dynamic mount points for hosts.

As shared storage subsystems vary in their handling of fencing, it is the responsibility of the hardware partner and their storage partners to develop a corruption-safe failover solution that is specific for the file system used to access that storage subsystem. An NFSv3 storage solution must be used in combination with the storage connector supplied by the hardware partner. NFSv4 and GPFS storage solutions can optionally be used with a storage connector.

A shared storage system could be configured as in the following figure. However, mounts may differ among hardware partners and their configurations.

High Availability Using Non-shared Storage

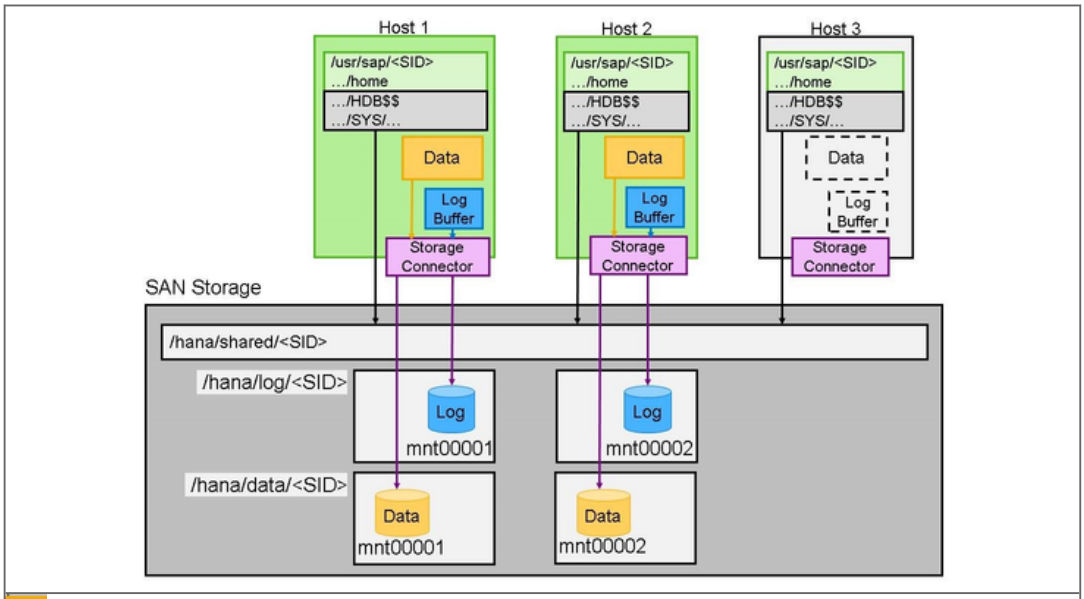


Figure 18: Non-shared Storage

It is also possible to assign separate storage to every SAP HANA host, which has nothing mounted except the shared area. An SAN storage must be used in combination with the SAP Fiber Channel Storage Connector, which SAP HANA offers to storage technology vendors. During failover, SAP HANA uses the storage connector API to tell the storage device driver to re-mount the required data and log volumes to the standby host, and fence off the same volumes from the failed host.

In a non-shared environment, separate storage is used in combination with the storage connector API. For more information about the storage connector API, see the SAP Fiber Channel Storage Connector Admin Guide available in SAP Note: 1900823.

Adding a Host to a Single-Host System

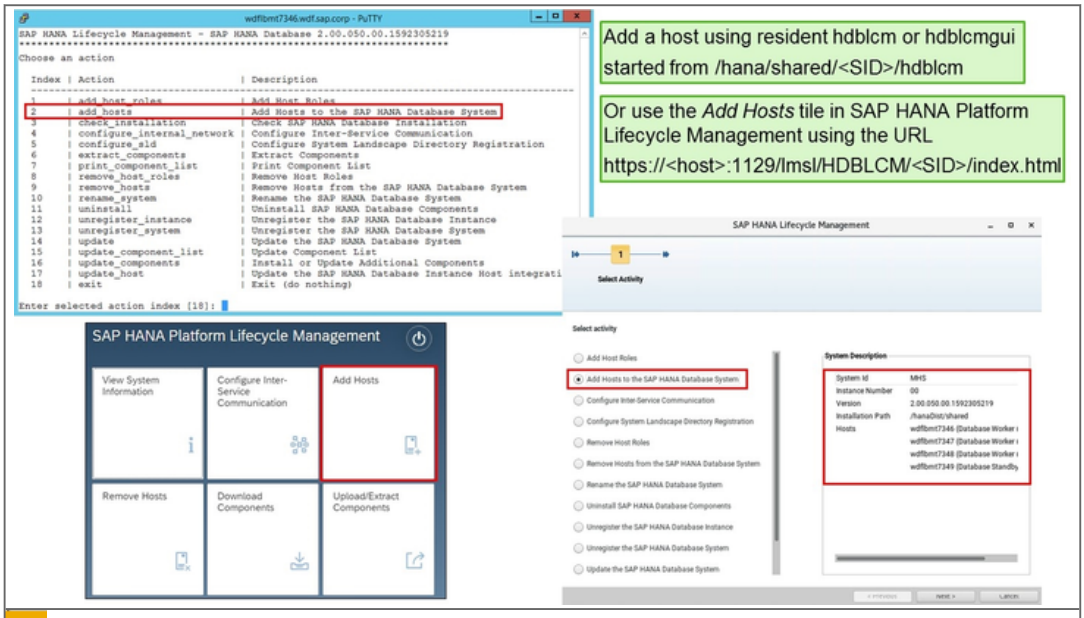


Figure 19: Use Resident SAP HANA Database Lifecycle Manager to Add a Host

An SAP HANA system can be configured as a multiple-host system during installation, using the SAP HANA database lifecycle manager (HDBLCM) from the installation media. It is also possible to add hosts after the installation of a single-host or multiple-host SAP HANA system, using the resident SAP HANA database lifecycle manager.

Use the SAP HANA database lifecycle manager graphical user interface, the command-line interface, or the SAP HANA database lifecycle manager Web user interface, to add one or multiple hosts to an existing SAP HANA system. The configuration options change depending on how the host is added.

Adding Hosts from an Integrated Host

The first consideration is whether the host you are logged on to is integrated in the system. If you are logged on to a configured system host, then you are on an integrated host and adding a non-integrated host to the system. In the following figure, the hosts in the dotted line (hanahost1 and hanahost2) are integrated hosts because they both belong to the SAP HANA system DB1. Consider being logged on to hanahost1, and adding the non-integrated host, hanahost3, to the SAP HANA system. The SAP HANA database lifecycle manager is started on the integrated host, hanahost1, and the addhost configuration task is carried out. The host information for hanahost3 is entered, and hanahost3 is configured as either a worker host or a standby host. As soon as the addhost configuration task is finished, hanahost3 has access to the shared storage of the DB1 system. It is also possible to add multiple non-integrated hosts to the same system at one time.

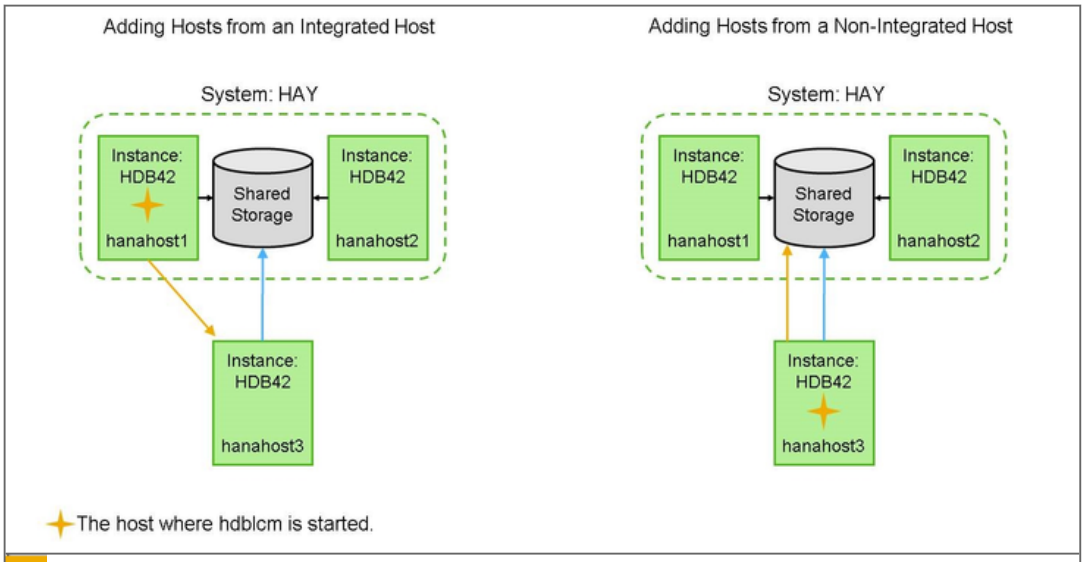


Figure 20: Adding a Host Integrated or Non-Integrated

Adding Hosts from a Non-Integrated Host

Alternatively, a non-integrated host can add itself to an SAP HANA system. This is referred to as adding a host from a non-integrated host, because you are logged on to a host that you want to add to the system.

To add multiple hosts to an SAP HANA system from a non-integrated host, first the non-integrated host must be added (and, therefore, become integrated), and then it can add more hosts. The SAP HANA database lifecycle manager interface is designed so that the non-integrated host and the additional hosts can be added in the same procedure. In the following figure, the non-integrated host has already been newly added to the system (become integrated), and is now adding the other hosts.

If you are adding a host to a single-host system, the listen interface is automatically configured to global during the host addition. After the host is added to the system, the internal network address can be defined and the inter-service communication can be re-configured to a different setting, if required.

Add Host Prerequisites

The following prerequisites must be fulfilled before hosts can be added to a SAP HANA system:

- 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.
- 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. Ensure 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 resident program does not modify the properties of any existing user or group.

The Exercise Explained

The following exercise demonstrates the installation of a four-node SAP HANA scale-out system. The original system was installed as a single-host system, with the data and log volumes stored on shared storage. This single-host system is extended to a four-node SAP HANA scale-out system by adding two worker nodes and one standby node.

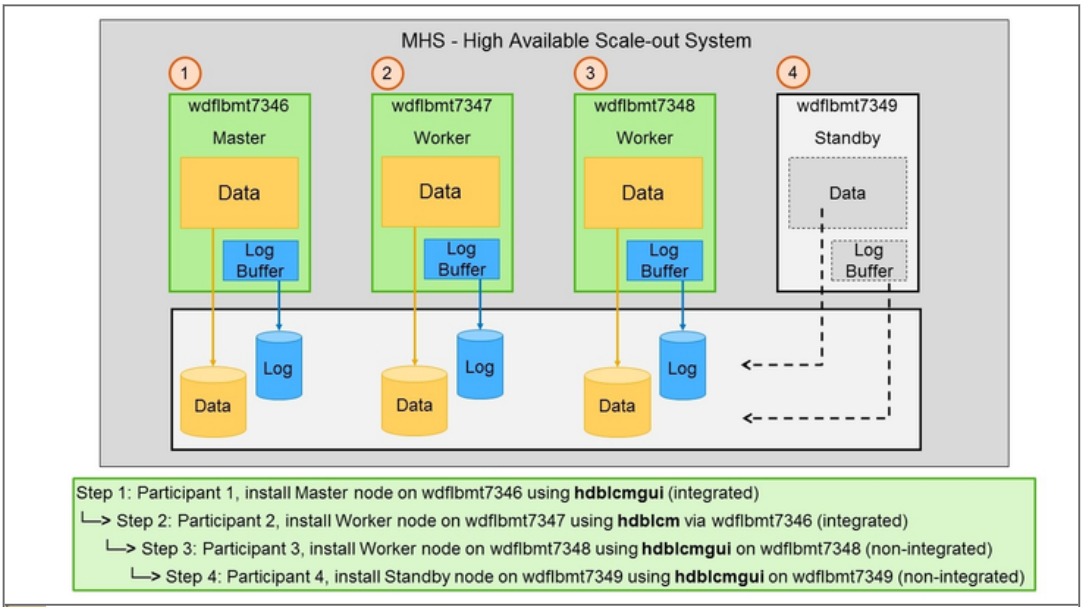


Figure 21: The Exercise Explained

1. In Step 1, the host wdflbmt7346 is installed as a single-host system.
2. In Step 2, the host wdflbmt7347 is added to the system, as a worker node, from the integrated host wdflbmt7346.
3. In Step 3, the host wdflbmt7348 is added to the system, as a worker node, from the non-integrated host wdflbmt7348.
4. In Step 4, the host wdflbmt7349 is added to the system, as a standby node, from the non-integrated host wdflbmt7349.

Related Information

SAP Note: 1900823 - SAP HANA Storage Connector API

SAP Note: 405827 - Linux: Recommended file systems

SAP Note: 2453736 - How-To: Configuring SAP HANA for SAP BW Extension Node in SAP HANA 2.0

LESSON SUMMARY

You should now be able to:

Install a high availability SAP HANA system

Unit 2
Lesson 2

Explaining SAP HANA Scale-Out

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Introducing SAP HANA scale-out systems

Introduction to SAP HANA Scale-Out Systems

Business Example

As an SAP HANA database administrator, you need to install and administrate your company's high availability scale-out SAP HANA systems. You need to understand the basic concept behind the SAP HANA scale-out technology.

Scaling SAP HANA

Scale-up and scale-out are the two general approaches you can take to enlarge your SAP HANA system.

Scale-up means increasing the size of one physical machine by increasing the amount of RAM available for processing.

Scale-out means combining multiple independent computers into one system. The main reason for distributing a system across multiple hosts (that is, scaling out) is to overcome the hardware limitations of a single physical server. This allows an SAP HANA system to distribute the load between multiple servers. In a distributed system, each index server is usually assigned to its own host to achieve maximum performance. It is possible to assign different tables to different hosts (partitioning the database), as well as to split a single table between hosts (partitioning of tables).

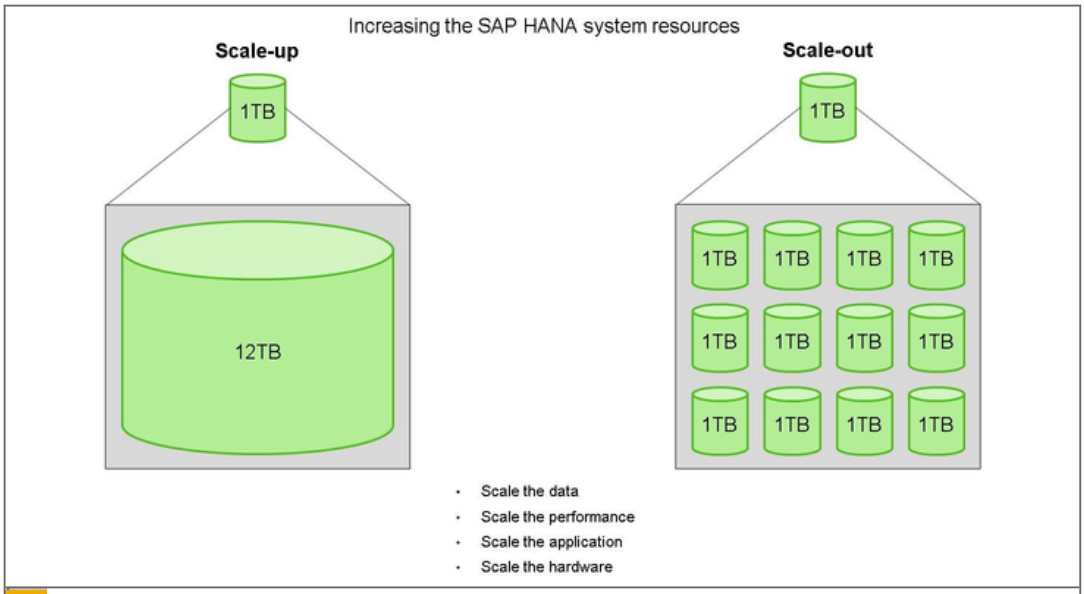


Figure 22: Scale-up vs Scale-out

Aspects of Scalability

You can use an SAP HANA scale-out architecture to manage large amounts of data or for higher availability. If you need to use more memory or more CPU power beyond the limitation of a single physical hardware box, you can use a distributed landscape consisting of multiple hosts.

Before you decide how to scale your SAP HANA implementation, there are several aspects that need to be considered, such as scaling data, performance, applications, and hardware.

Scaling the Data

One technique you can use to deal with planned data growth is to purchase more physical RAM than is initially required, to set the allocation limit according to your needs, and then to increase it over time to adapt to your data. Once you have reached the physical limits of a single server, you can scale out over multiple machines to create a distributed SAP HANA system. You can do this by distributing different schemata and tables to different servers (complete data and user separation). However, this is not always possible, for example, when a single fact table is larger than the server's RAM size.

The most important strategy for scaling your data is data partitioning. Partitioning supports the creation of very large tables (billions of rows) by breaking them into smaller chunks that can be placed on different machines. Partitioning is transparent for most SQL queries and other data manipulations.

Scaling Performance

SAP HANA’s performance is derived from its efficient, parallel approach. The more computation cores your SAP HANA server has, the better the overall system performance.

Scaling performance requires a more detailed understanding of your workload and performance expectations. Using simulations and estimations of your typical query workloads, you can determine the expected load that a typical SAP HANA installation may comfortably manage. At the workload level, a rough prediction of scalability can be established by measuring the average CPU utilization while the workload is running. For example, an average CPU utilization of 45% may indicate that the system can be loaded 2X before showing a significant reduction in individual query response time.

Scaling the Application

Partitioning can be used to scale the application as it supports an increasing number of concurrent sessions and complex analytical queries by spreading the calculations across multiple hosts. Particular care must be taken in distributing the data so that the majority of queries match partitioning pruning rules. This accomplishes two goals: directing different users to different hosts (load balancing) and avoiding the network overhead related to frequent data joins across hosts.

Scaling Hardware

SAP HANA is offered in a number of ways – in the form of an on-premise appliance, delivered in a number of different configurations and "sizes" by certified hardware partners or by using the tailored data center integration model, and as part of a cloud-based service. This creates different system design options with respect to scale-up and scale-out variations. To maximize performance and throughput, SAP recommends that you scale up as far as possible (acquire the configuration with the highest processor and memory specification for the application workload), before scaling out (for deployments with even greater data volume requirements).

Note:
SAP HANA hardware partners have different building blocks for their scale-out implementations. Therefore, you should always consult with your hardware partner when planning your scale-out strategy.

Introducing High-Availability in an SAP HANA system

In the figure Scale-up vs Scale-out, the SAP HANA systems are only increased in size from 1TB to 12TB. In both scenarios, scale-up and scale-out, no high availability is introduced. High availability can only be introduced in a scale-out setup with inclusion of standby nodes. A scale-out configuration with high availability is shown in the figure High Availability and Scale-out. One or more SAP HANA nodes can be configured as standby nodes. A standby node automatically takes over the operations of a failed host using the host auto-failover feature from SAP HANA.

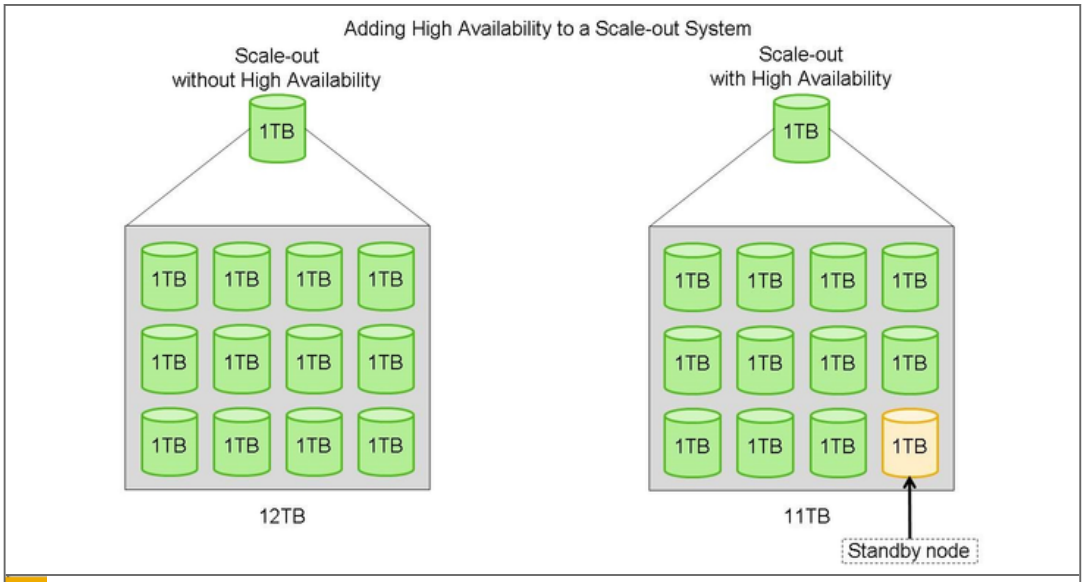


Figure 23: High Availability and Scale-out

As soon as you introduce standby nodes in an SAP HANA scale-out configuration, you reserve resources for the event of a failure. These resources cannot be used in the active system. In the figure High Availability and Scale-out, one host is defined as the standby node. This means that our system now has only 11 nodes active nodes, and the total database size is reduced from 12TB to 11TB.

A scale-up configuration, by default, has no high availability capabilities. This is due to the fact that a scale-up system consists of one server. A scale-up system can be made high availability by adding an additional standby host to the SAP HANA system, or by setting up a configuration using storage replication or system replication.

Multiple-host (Distributed) Systems

An SAP HANA system can comprise multiple isolated databases and may consist of a cluster of several hosts. This is referred to as a multiple-host, distributed system, or scale-out system, and supports scalability and availability.

An SAP HANA system is identified by a single system ID (SID) and contains one or more tenant databases and one system database. Databases are identified by an SID and a database name. From the administration perspective, there is a distinction between tasks performed at the system level and those performed at the database level. Database clients, such as the SAP HANA cockpit, connect to specific databases.

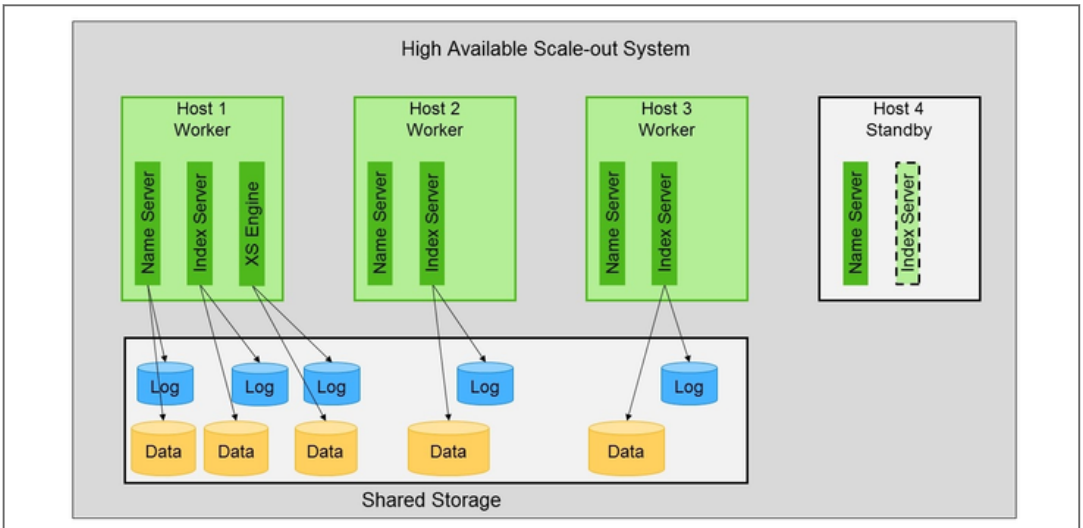


Figure 24: High Available Scale-out System

A host is a machine that runs parts of the SAP HANA system. The machine is comprised of CPU, memory, storage, network, and an operating system.

An SAP HANA instance is the set of components of a distributed system that are installed on one host. The figure High Available Scale-out System shows a distributed system that runs on four hosts. In this example, each instance has an index server and a name server.

One or more hosts can be configured to work in standby mode, so that if an active host fails, a standby host automatically takes its place. The index servers on standby hosts do not contain any data and do not receive any requests.

The index server contains all the database and processing components. Each index server is a separate operating system process and it also has its own disk volumes. When processing database operations, index servers may need to forward the execution of some operations to other servers that own the data involved in the operation.

In each SAP HANA system, there is one master index server. It stores the meta-data and it contains the master transaction manager that coordinates distributed transactions involving multiple index servers.

Database clients can send their requests to any index server. If the contacted index server does not own all the data involved, it delegates the execution of some operations to other index servers, collects the result, and returns it to the database client.

In a distributed system, a central component is required that knows the topology and how data is distributed. This component is the name server. The name server knows which tables, table replicas, or partitions of tables are located on which index server.

When processing a query, the index servers ask the name server about the locations of the involved data. To prevent a negative impact on performance, the topology and distribution information is replicated and cached on each host. In each SAP HANA system, there is one master name server that owns the topology and distribution data. This data is replicated to all other name servers, called slave name servers. The slave name servers write the replicated data to a cache in shared memory from where the index servers of the same instance can read it.

The master name server has its own persistence where it stores name server data (topology and distribution data). The slave name servers have no persistence because they are only holding replicated data.

LESSON SUMMARY
You should now be able to:

- Introducing SAP HANA scale-out systems

Unit 2
Lesson 3

Partitioning Tables

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Perform table partitioning tasks

Table Partitioning

Data Distribution in SAP HANA

In a multiple-host system, each index server is usually assigned to its own host for maximum performance. SAP HANA supports the following different ways of distributing data across the hosts:

- A partitioned table splits its data in several blocks (partitions), and these partitions can be stored on different index servers.
- Different tables can be assigned to different index servers.
- A table can be replicated to multiple index servers, for better query and join performance.

When you create new tables, or partitions, they are distributed to the available hosts by the system. By default, a “round-robin” distribution method is used, but tables can also be positioned by using the table placement rules or by specifying a host and port number with the SQL CREATE TABLE statement in the location clause. This gives the developer complete control over the positioning of individual tables.

Specific applications may have predefined table distribution rules, and in some cases, configuration files and documentation are available in SAP Notes to help you to set up the necessary partitioning and table placement rules.

Introduction to Table Partitioning

The partitioning feature of the SAP HANA database splits column-store tables horizontally into disjunctive sub-tables or partitions. In this way, large tables can be broken down into smaller, more manageable parts.

Partitioning is only available for tables located in the column store. The row store does not support partitioning.

Hint:
Partitioning is typically used in multiple-host systems, but it may also be beneficial in single-host systems.

Table partitioning is transparent for the application in a way that applications work properly with all partitioning strategies. Nevertheless, partitioning can have an impact on performance, so it can make a difference for the end user and the system load, both in a positive and

negative way. To minimize the risk of performance regressions, it is important to implement a good partitioning strategy.

Partitioning is transparent for SQL queries and data manipulation language (DML) statements. The following additional data definition statements (DDL) for partitioning are available:

- Perform the delta merge operation on certain partitions
- Create table partitions
- Re-partition tables
- Merge partitions to one table
- Add or delete partitions
- Move partitions to other hosts

> Note:
> After adding or removing hosts, it is recommended that you execute a redistribution operation. Based on its configuration, the redistribution operation suggests a new placement for tables and partitions in the system. If you confirm the redistribution plan, the redistribution operation redistributes the tables and partitions accordingly.

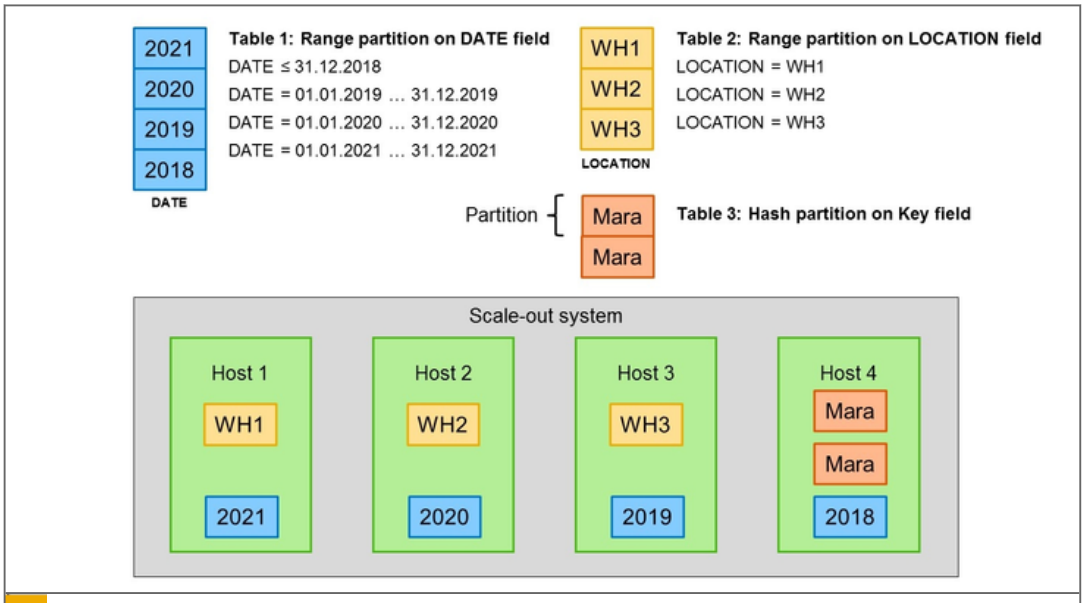


Figure 25: Table Partitioning

When a table is partitioned, the split is done in such a way that each partition contains a different set of rows of the table. There are several alternatives available for specifying how the rows are assigned to the partitions of a table, for example, hash partitioning, round-robin partitioning, or partitioning by range.

Hash Partitioning

Hash partitioning is used to distribute rows to partitions equally for load balancing and to overcome the 2 billion row limitation. The number of the assigned partition is computed by

applying a hash function to the value of a specified column. Hash partitioning does not require an in-depth knowledge of the actual content of the table.

For each hash partitioning specification, columns must be specified as partitioning columns. The actual values of these columns are used when the hash value is determined. If the table has a primary key, these partitioning columns must be part of the key. The advantage of this restriction is that a uniqueness check of the key can be performed on the local server. You can use as many partitioning columns as required to achieve a good variety of values for an equal distribution.

For more information about the SQL syntax for partitioning, see SAP HANA SQL and System Views Reference.

Round-Robin Partitioning

Round-robin partitioning is used to achieve an equal distribution of rows to partitions. However, unlike hash partitioning, you do not have to specify partitioning columns. With round-robin partitioning, new rows are assigned to partitions on a rotation basis. The table must not have primary keys.

Hash partitioning is usually more beneficial than round-robin partitioning for the following reasons:

- The partitioning columns cannot be evaluated in a pruning step. Therefore, all partitions are considered in searches and other database operations.
- Depending on the scenario, it is possible that the data in semantically-related tables resides on the same server. Some internal operations may then operate locally instead of retrieving data from a different server.

Range Partitioning

Range partitioning creates dedicated partitions for certain values or value ranges in a table. For example, a range partitioning scheme can be chosen to create a partition for each calendar month. Partitioning requires an in-depth knowledge of the values that are used or are valid for the chosen partitioning column.

Partitions may be created or dropped as needed and applications may choose to use range partitioning to manage data at a fine level of detail, for example, an application may create a partition for an upcoming month so that new data is inserted into that new partition.

Note:
Range partitioning is not well suited for load distribution. Multi-level partitioning specifications address this issue.

When rows are inserted or modified, the target partition is determined by the defined ranges. If a value does not fit into one of these ranges, an error is raised. To prevent this, you can also define an 'others' partition for any values that do not match any of the defined ranges. The 'others' partitions can be created or dropped on-the-fly as required.

Range partitioning is similar to hash partitioning in that the partitioning column must be part of the primary key. Many data types are supported for range partitioning. See the list of data types in Partitioning Limits for the complete list.

Multi-Level Partitioning

Multi-level partitioning can be used to overcome the limitation of single-level hash partitioning and range partitioning, that is, the limitation of only being able to use key columns as

partitioning columns. Multi-level partitioning makes it possible to partition by a column that is not part of the primary key.

Explicit Partition Handling for Range Partitioning

For all partitioning specifications involving range, it is possible to have additional ranges added and removed as necessary. This means that partitions are created and dropped as required by the ranges in use. In the case of multi-level partitioning, the desired operation is applied to all relevant nodes.

Note:
If a partition is created and an others partition exists, the rows in the others partition that match the newly-added range are moved to the new partition. If the others partition is large, this operation may take a long time. If an others partition does not exist, this operation is fast as only a new partition is added to the catalog.

Range partitioning requires at least one range to be specified regardless of whether or not there is an others partition. When partitions are dropped, the last partition created cannot be dropped even if an others partition exists.

For range-range partitioning you have to specify whether a partition must be added or dropped on the first or second level by specifying the partitioning column.

Caution:
The DROP PARTITION command deletes data. It does not move data to the others partition.

Time Selection Partitioning (Aging)

The SAP HANA database offers a special time selection partitioning scheme, also called "aging". Time selection or aging allows SAP Business Suite application data to be horizontally partitioned into different temperatures like hot and cold.

SAP Business Suite ABAP applications can use aging, which must not be used for customer or partner applications, to separate hot (current) data from cold (old) data by using time selection partitioning to:

- Create partitions and re-partition
- Add partitions
- Allocate rows to partitions
- Set the scope of Data Manipulation Language (DML) and Data Query Language (DQL) statements

Setting the DML and DQL scope is the most important aspect of time selection partitioning. It uses a date to control how many partitions are considered during SELECT, CALL, UPDATE, UPSERT, and DELETE. This date may be provided by the application with a syntax clause and it restricts the number of partitions that are considered.

Caution:
Tables with time selection partitioning cannot be converted into any other kind of tables using ALTER TABLE .

Partitioning Advantages

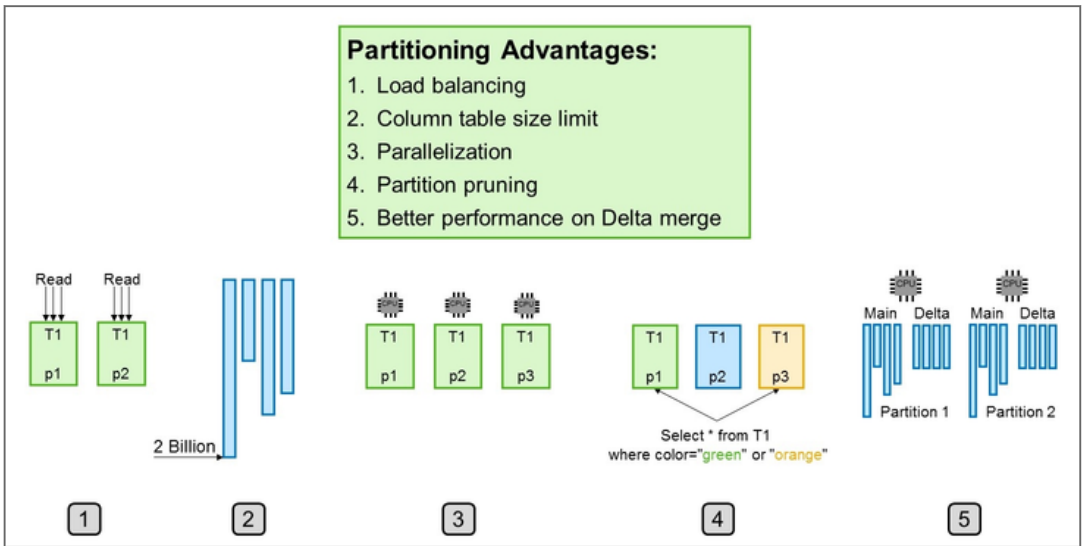


Figure 26: Advantages of Table Partitioning

Load balancing in a distributed system

Individual partitions can be distributed across multiple hosts. This means that a query on a table is not necessarily processed by a single server, but may be processed by all the servers that hold one or more partitions of that table.

Overcoming the size limitation of column-store tables

A non-partitioned table cannot store more than 2 billion rows. It is possible to overcome this limitation by distributing the rows across several partitions. Each partition can not contain more than 2 billion rows.

Parallelization

Partitioning allows operations to be parallelized by using several execution threads for each table.

Partition pruning

Queries are analyzed to determine whether or not they match the given partitioning specification of a table (static partition pruning) or match the content of specific columns in aging tables (dynamic partition pruning). If a match is found, it is possible to determine the specific partition that holds the data being requested, and accordingly avoiding the access and loading of partitions that are not required into memory.

Improved performance of the delta merge operation

During a delta merge operation of an unpartitioned table, the entire table must be duplicated during the merge operation. This requires a large amount of RAM.

During the delta merge operation of a partitioned table, only modified partitions are subject to the merge operation. This requires less RAM because not every partition has been changed.

> Caution:
> Before a table is partitioned or re-partitioned, a delta merge operation is executed. Therefore, in the case of huge tables, you must partition them in good time so as not to run out of memory during the merge operation.

Explicit partition handling

In some cases, it can be useful that the application controls the creation and existence of partitions based on specific criteria, for example, by adding partitions to store the data for an upcoming month.

Examples of Using Table Partitioning with SQL

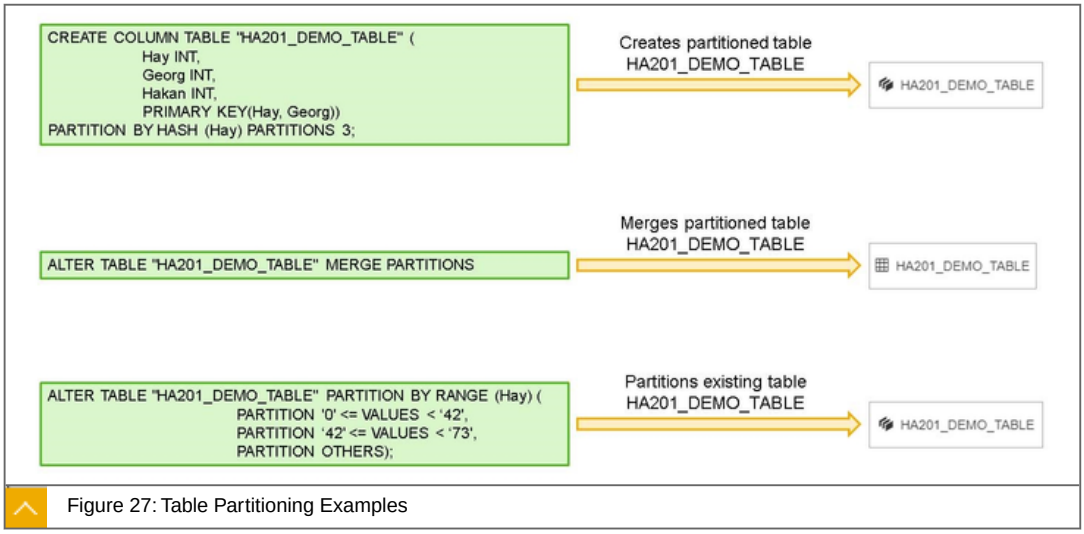


Figure 27: Table Partitioning Examples

In the figure, Table Partitioning Examples, the table HA201_DEMO_TABLE is created with three partitions. The single-level partitioning specification is HASH on column Hay.

The above example creates a new table that is partitioned during creation. In real life, it will also happen that you need to merge one or more partitioned tables, or partition an existing table. The use cases may be that the table is nearing the 2 billion records limit, or the query performance is not optimal. Using the mentioned SQL statements, you can merge partitioned tables, or partition existing tables in the SAP HANA database.

Repartitioning

There is no automatic repartitioning when threshold values are exceeded. Instead, this is proposed the next time the redistribution process is executed.

The values entered for partitioning must be consistent with the physical landscape, especially the number of server nodes available.

If repartitioning is necessary, tables are only repartitioned by doubling the number of existing (initial) partitions. This is done for performance reasons. The maximum number of (first-level) partitions reached by that process is defined by parameter global.ini > [table_placement] > max_partitions (default: 12).

By default, the system does not create more partitions than the number of available hosts (or more specifically, possible locations). For example, if INITIAL_PARTITIONS is set to 3, but the distributed SAP HANA database has five possible locations, repartitioning from three to six partitions does not take place. A table can have more than one partition per host if the parameter global.ini > [table_placement] > max_partitions_limited_by_locations is set to false (default: true). This rule is disregarded if a higher number of first level partitions is required to partition groups with more than 2 billion records (global.ini > [table_placement] > max_rows_per_partition , default = 2,000,000,000).

Note:
SAP Note: 2044468 - “FAQ: SAP HANA Partitioning” provides detailed information on SAP HANA partitioning.

Table Distribution Editor: Additional Actions

If a table is distributed to several partitions, it displays the host that stores each of these partitions. Existing partitions can be moved to different hosts by generating a specific redistribution plan. You can also balance table distribution after adding new hosts to the system. Check, optimize, compress, defrag, load table, delta merge, and evaluate repartitioning of tables that are not partitioned to other hosts as well.

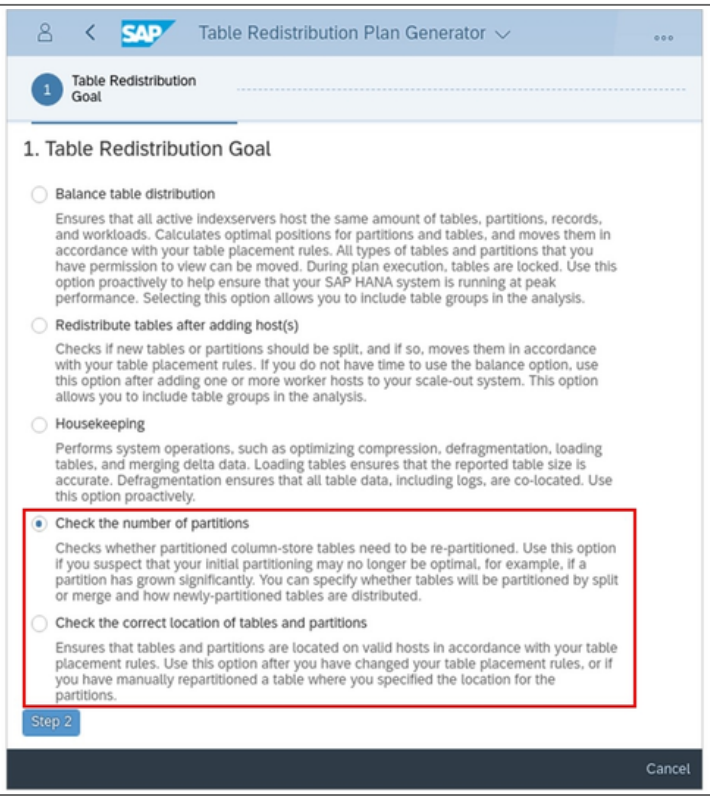


Figure 28: Generate Redistribution Plan

Note:
Before moving tables or partitions, the system checks that the host has sufficient memory.

Changing how tables are distributed across hosts is a critical operation. Back up the landscape before executing a redistribution operation.

Best Practices for Table Partitioning

To create an optimal partitioning plan, you should try to follow the table partitioning best practices in the following figure.

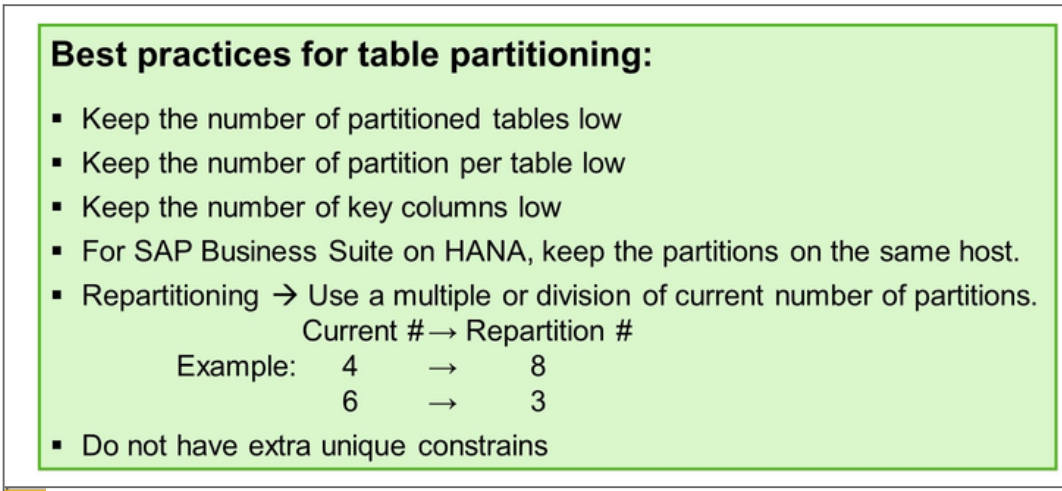


Figure 29: Best Practices Table Partitioning

Keep the number of partitioned tables low

Only partition tables if you see a clear benefit without significant regressions.

Keep the number of partitions per table low

An unnecessarily high amount of partitions result in overhead because some queries may have to access all partitions to find the data:

- A high amount of network channels are opened and so the system is at risk of reaching the max channels limitation (SAP Note: 2222200) and running into network-related terminations.
- Certain operations like the determination of column statistics (SAP Note: 2114710) have to be performed individually for each partition.

So, consider the following general rules before defining a certain number of partitions:

- If you partition tables due to the 2 billion limit, it is usually acceptable if individual partitions contain up to 1.5 billion records (less if you expect a significant future growth).
- If you partition by date, you should avoid using granular ranges (such as days or weeks) resulting in a high amount of partitions.

- If you use a RANGE partition on columns with data that is not evenly distributed (such as a number range column with multiple different number ranges), you should check the actual value distribution and define the range limits accordingly.

Keep the number of key columns low

As few partition key columns as possible. It is useful to keep the number of partition key columns to a minimum for the following reasons:

- Partition pruning of hash partitions can only be used if all underlying partitioning columns are specified with "=" or "IN" in the WHERE clause.
- Determining partition pruning can be quite time consuming if many partition keys are involved.

  For more information, see SAP Note: 2000002 "What are typical approaches to tune expensive SQL statements?" and "Execution time higher than expected, negative impact by existing partitioning".

In the case of hash partitioning, it is often useful to use only the most selective primary key column as the partition key column.

For SAP Suite on HANA, keep all partitions on same host

In scale-out SAP Suite on HANA environments, it is advantageous to keep all partitions of a table on the same host. As of SPS08, this can achieved with an appropriate table placement configuration.

As a fallback option, you can use a dummy first-level partitioning (for example, on MANDT) and perform the actual partitioning at the second level. In this case, all partitions are located on the same host.

Repartitioning rules

When repartitioning, choose the new number of partitions as a multiple or divider of current number of partitions.

If a table is already partitioned, it is most efficient to choose a new number of partitions that is a factor 2 multiple or divider of the current number of partitions (such as 4 -> 8 or 6 -> 3 partitions). Only in this case can the repartitioning happen in parallel on different partition groups and hosts (“parallel split/merge”).

Avoid unique constraints

When creating partitions, try to avoid creating additional unique constraints. Avoid partitioning tables with additional unique constraints (such as a unique secondary index), as the uniqueness checks impose significant overhead.

Note:
SAP Note: 2000002 gives insight into SAP HANA SQL optimization, and describes symptoms that can be introduced by inadequate partitioning.

Table Partitioning Monitoring Views

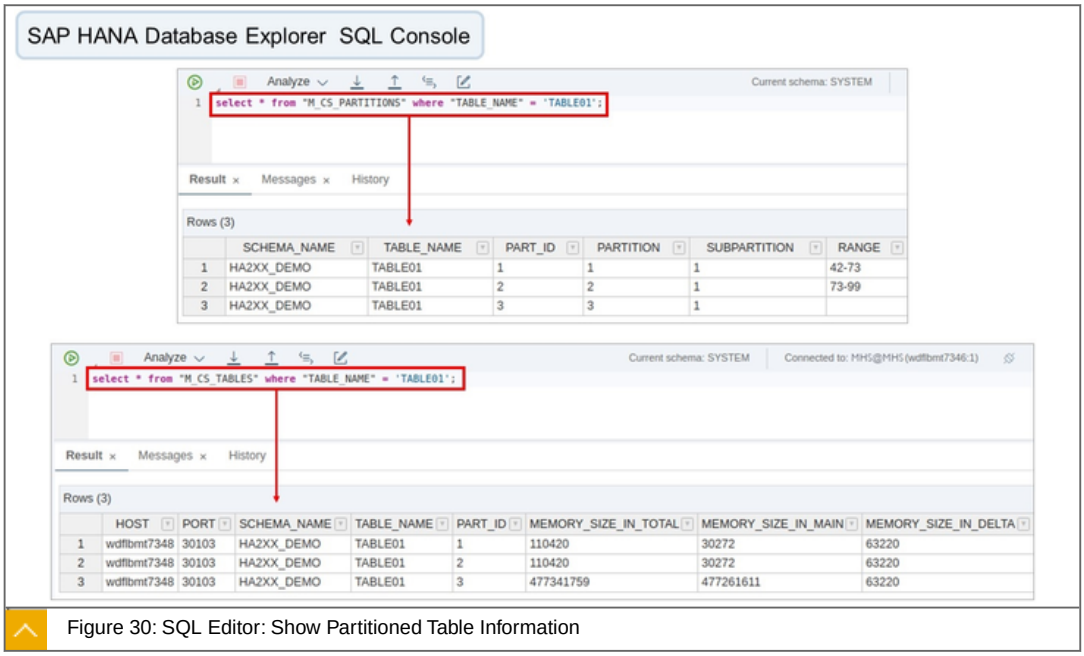


Figure 30: SQL Editor: Show Partitioned Table Information

The M_CS_PARTITIONSsystem view provides partition information of column tables.

```
select * from "M_CS_PARTITIONS" where "TABLE_NAME" =
'sap.hana.democontent.epm.data::PO.Item';
```

Results in:

The output shows the number of partitions. In the above example, we have three partitions.

The M_CS_TABLESsystem view provides run time data for column tables or partitions of column tables.

```
select * from "M_CS_PARTITIONS" where "TABLE_NAME" =
'sap.hana.democontent.epm.data::PO.Item';
```

The output shows which host the partition is located on, and how much memory is consumed by the table.

The M_EFFECTIVE_TABLE_PLACEMENTsystem view provides information on the table placement location. This view also contains information about the partitioning thresholds. You can see the valid location(s) according to the configuration, the actual values for each partitioning parameter, and in the corresponding _MATCH columns, the reason (matching rule) for those.

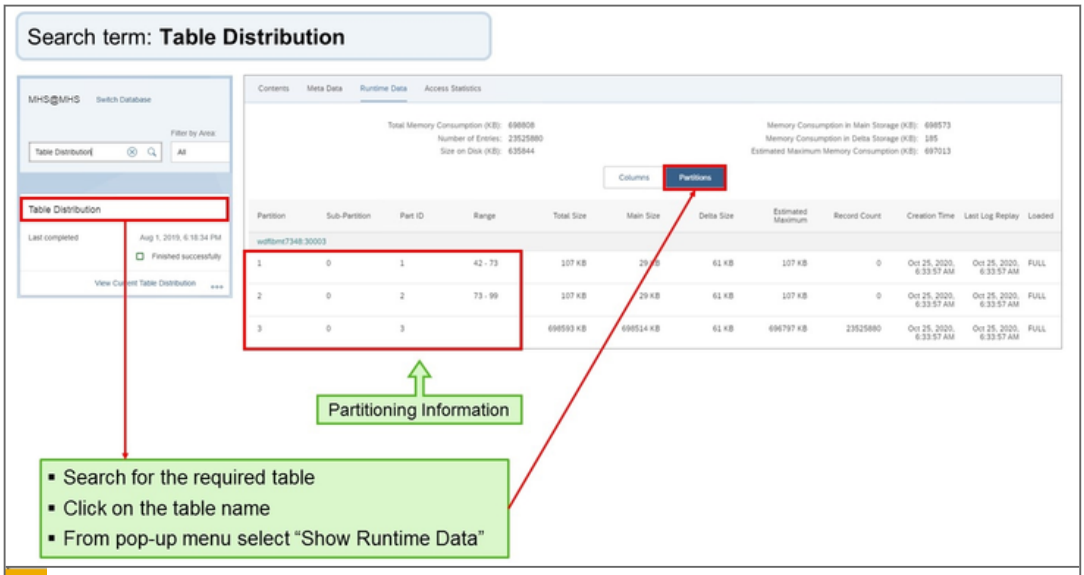


Figure 31: HANA Cockpit: Show Partitioned Table Information

The table information is also available in SAP HANA Cockpit. In the View Current Table Distribution application, search for the required table and select it. In the pop-up, select the Show Runtime Data option. On the Runtime Data screen, the table definition is shown by default. Select the Partitions button to get an overview of how the table is partitioned and where the partitions are located. The partition range is also displayed.

Table Consistency Checks

To ensure consistency for partitioned tables, execute checks and repair statements, if required.

You can call general and data consistency checks for partitioned tables to check, for example, that the partition specification, metadata, and topology are correct.

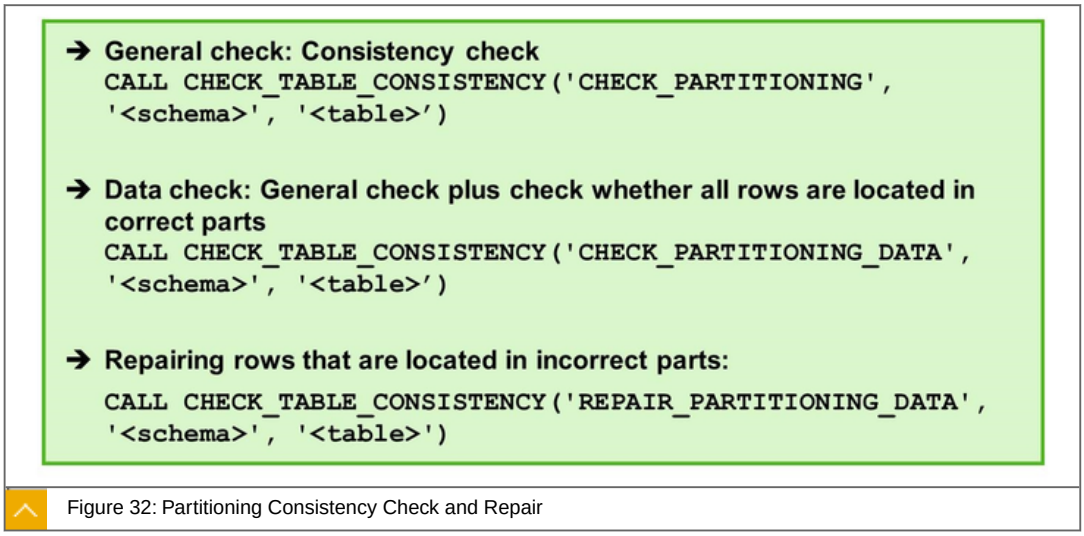


Figure 32: Partitioning Consistency Check and Repair

Note:
The data checks can take a long time to run, depending on the data volume.

LESSON SUMMARY
You should now be able to:

- Perform table partitioning tasks

Unit 2
Lesson 4

Table Placement

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Perform Table Placement Tasks

Table Placement

Table Placement

Table classification and table placement configuration, enhanced by partitioning, build the foundation for controlling the data distribution in an SAP HANA scale-out environment.

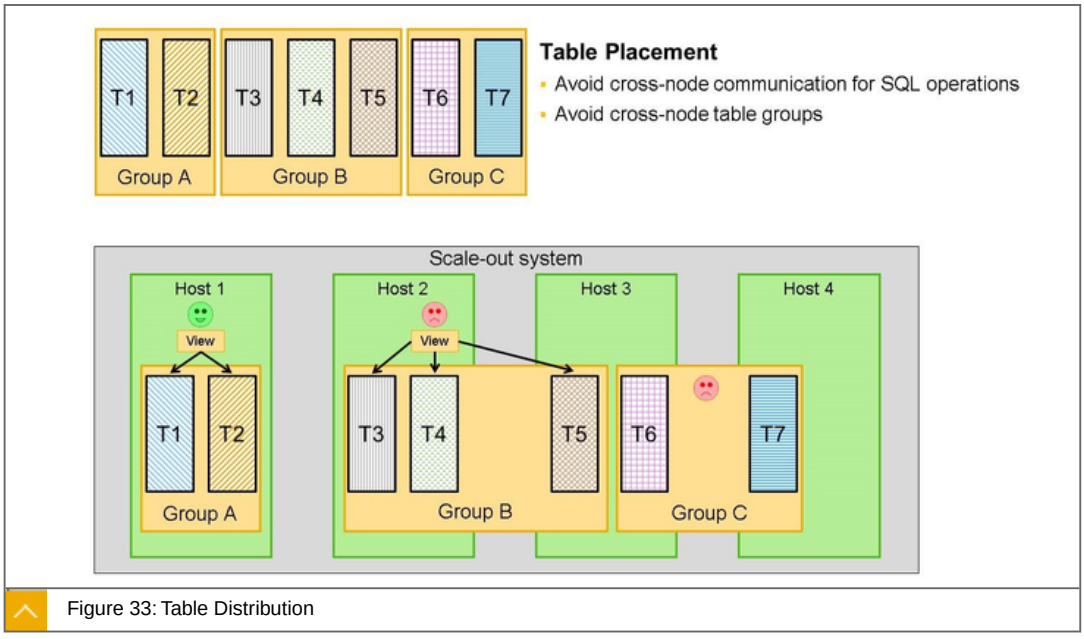


Figure 33: Table Distribution

Associated tables can be classified by a common table group.

The SQL interface of SAP HANA provides three possible kinds of classifications: group name, group type, and subtype. Tables that have been classified with group information are included in the SYS.TABLE_GROUPS table, and you can review classification details in the monitoring view SYS.TABLE_GROUPS (see details following). Tables with the same group name are kept on the same host, or, in the case of partitioned tables that are distributed over several hosts, corresponding first-level partitions are distributed for all tables the same way.

One table in the group is defined as the leading table and its table placement settings are applied to all other tables in the group. This can be, for example, the location, or in the case of partitioned tables (if SAME_PARTITION_COUNT is set in SYS.TABLE_PLACEMENT, see below), the number of first-level partitions.

Note:
Specific applications, such as SAP BW, classify objects automatically as they are created. These classifications must not be changed manually.

For native applications, the application developer can define a grouping manually, for example, by grouping tables together that are often joined. The following statements show examples of setting table group attributes using CREATE and ALTER statements:

```
CREATE COLUMN TABLE "HA201_DEMO"."HA201_TABLE"(
 HAY INT,
 GEORG INT,
 HAKAN INT,
 PRIMARY KEY (HAY, GEORG))
GROUP NAME DEMO GROUP TYPE EXAMPLE GROUP SUBTYPE TEST GROUP LEAD;
```

This create table statement creates a table named HA201_TABLE, sets the group name to DEMO, the group type to EXAMPLE, the group subtype to TEST, and makes this the lead table in the group DEMO.

```
ALTER TABLE "HA201_DEMO"."HA201_TABLE" SET
 GROUP NAME DEMO
 GROUP TYPE EXAMPLE
  GROUP SUBTYPE TEST
 GROUP LEAD;
```

This alter table statement sets the group name to DEMO, the group type to EXAMPLE, the group subtype to TEST, and makes this the lead table in the group DEMO, for an existing table.

This can also be performed dynamically based on the information available in the SQL Plan Cache, with the Join Path Analysis tool within the Data Distribution Optimizer, or with the ABAP grouping report (SHDBSO_TABLE_GROUPING) for SAP S/4HANA scale-out. See SAP Note: 2447004 - “Table Grouping Report for S/4 HANA in scale-out systems”.

The SAP HANA Cockpit Table Redistribution tools also include an optional preparation step to integrate the Group Advisor tool into the plan generation process to create table groups dynamically.

Table Classification and Placement

Application data is usually stored in a multitude of database tables, and data from several of these tables is combined using SQL operations, such as join or union, when it is queried. As these relations between different tables are defined in the application code, this information is not available in SAP HANA. The table classification feature provides a possibility to push down this semantic information in the database by allowing administrators to define groups of tables. This information can be used, for example, when determining the number of partitions to be created, or, in the case of a scale-out landscape, the node on which to locate the tables or partitions.

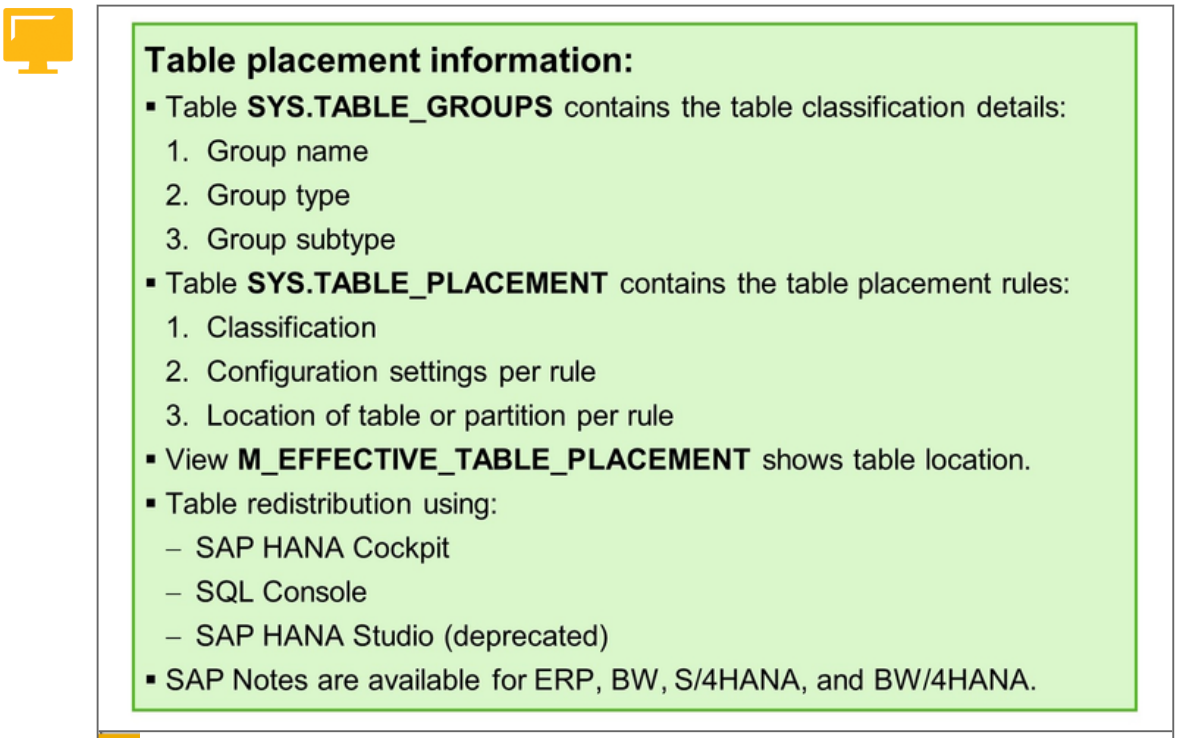


Figure 34: Table Placement Information

The classification is performed by providing each table with a group name, group type, and subtype. Based on combinations of these elements, as well as the table names and schema names, a set of configuration values can be defined as table placement rules. These rules are used to control, for example, the placement of partitions or the number of partitions during operations like table creation or redistribution. By doing this, associated or strongly-related tables are placed in such a way that the required cross-node communication is minimized for SQL operations on tables within the group.

Table placement rules are applied during system migration or table creation, but it may also be necessary to adjust the location or the number of partitions on an ongoing basis for handling data growth. Therefore, table redistribution can also be run on demand to optimize the landscape as the system evolves. Repartitioning is always necessary, for example, for any table or partition in the database that reaches the maximum count of 2 billion rows.

The following tools are available to perform table repartitioning and redistribution. These tools evaluate the current landscape and determine an optimized distribution:

SAP HANA Table Redistribution

Data Distribution Optimizer (part of SAP HANA Data Warehousing Foundation)

Balancing an SAP HANA scale-out landscape with these tools is done in two stages:

1. Generation of a plan based on table placement rules (described in detail in a later section). After generating the plan, you can review it and adjust the definition of the rules if required.
2. Execution of the plan that implements the partitioning and distribution changes.

Because split table and move table are operations that require table locks, the execution of the plan should not be performed during a period where there is heavy load on the database.

Table Classification and Table Placement Rules

Table placement rules are defined in the table SYS.TABLE_PLACEMENT. The system privilege TABLE ADMIN is required to maintain these settings. Placement rules basically address the following areas:

- Classification, that is, related tables that must be located together are organized in groups
- Configuration settings to manage partitioning (number of initial partitions, split threshold, and so on)
- Physical distribution or location of tables or partitions in the server landscape

When creating a table, if the defined rules match with a table or a table group, SAP HANA considers them while creating the table. Keep in mind that partition specifications must still be defined by the application.

Table Placement Rules

The TABLE_PLACEMENT table provides a customizing interface that can be used for the dynamic management of partitions and locations.

The partitioning parameters are used to define how a table or a group of tables is partitioned if the table has a first-level partitioning specification of hash or round-robin. Range partitioning is not handled in this way.

If the number of rows is lower than MIN_ROWS_FOR_PARTITIONING, the table consists of only one partition. If this minimum row limit is exceeded, the table is partitioned in as many parts as fulfills the following constraints:

- The number of partitions is larger or equal to the value of (row count of the table) / REPARTITIONING_THRESHOLD.
- The number of partitions is a multiple of INITIAL_PARTITIONS.
- The number of partitions is smaller or equal to the number of hosts if the parameter max_partitions_limited_by_locations is not set to false and the number of partitions is less than the value of the parameter max_partitions (see details below).

Therefore, if the table has more than one partition, there are at least INITIAL_PARTITIONS partitions, and each partition has less than REPARTITIONING_THRESHOLD records. In this context, partitions refer to first-level partitions (of type HASH or ROUNDROBIN).

Note that when a partitioned table is created without an estimated row count (default behavior), a partitioned table is created with INITIAL_PARTITIONS first-level partitions. Whereas in a redistribution, it is targeted to have a single first-level partition (assuming MIN_ROWS_FOR_PARTITIONING > 0). In specific applications, creation is performed with an estimated row count, for example, BW with 1 million, and therefore it is created with only one first level-partition (assuming MIN_ROWS_FOR_PARTITIONING > 1,000,000).

Repartitioning

There is no automatic repartitioning when threshold values are exceeded. Instead, this is proposed the next time the redistribution process is executed.

The values entered for partitioning must be consistent with the physical landscape, especially the number of server nodes available:

- If repartitioning is necessary, tables are only repartitioned by doubling the number of existing (initial) partitions. This is done for performance reasons. The maximum number of

(first-level) partitions reached by that process is defined by parameter global.ini > [table_placement] > max_partitions (default: 12).

By default, the system does not create more partitions than the number of available hosts (or more specifically possible locations). For example, if INITIAL_PARTITIONS is set to 3, but the distributed SAP HANA database has five possible locations, repartitioning from three to six partitions would not take place. A table can have more than one partition per host if the parameter global.ini > [table_placement] > max_partitions_limited_by_locations is set to false (default: true). This rule is disregarded if a higher number of first level partitions is required to partition groups with more than 2 billion records (global.ini > [table_placement] > max_rows_per_partition, default: 2,000,000,000).

Location

There are predefined values for possible locations:

master: represents the master node

slave (or slaves): represents all slave nodes that belong to the worker group ‘default’

all: represents all nodes that belong to the worker group ‘default’, that is, the master node and the slave nodes

The worker group assignment can be found in the WORKER_ACTUAL_GROUPS entry of the M_LANDSCAPE_HOST_CONFIGURATION view, and can be accessed by executing the following procedure:

```
call SYS.UPDATE_LANDSCAPE_CONFIGURATION('GET
WORKERGROUPS','<hostname>')
```

In addition, it is also possible to create custom location definitions by using the following procedure to assign worker groups to a host:

```
call SYS.UPDATE_LANDSCAPE_CONFIGURATION('SET
WORKERGROUPS','<hostname>','<name1> <name2> <name3>')
```

Note:
If a host is assigned to several worker groups, they must be separated by a space.

How Rules are Applied

The TABLE_PLACEMENT table is read in such a way that a more specific rule supersedes a more generic one. A complete matrix of priorities is available in SAP Note: 1908082 — “Table Placement Priorities”.

For example, an entry with only one schema applies to all tables of that schema; additional entries for that schema and specific group types overrule the more general rule.

Monitoring View

You can see the actual table placement settings per table by querying the M_EFFECTIVE_TABLE_PLACEMENT system view. You can see the valid location(s) according to the configuration, and for each partitioning parameter the actual values, and in the corresponding _MATCH columns the reason (matching rule) for those.

The information how a table is classified, can be reviewed in the SYS.TABLE_GROUPS monitoring view.

Redistributing Tables in a Multi-host SAP HANA System

In a distributed SAP HANA system, tables and table partitions are assigned to an index server on a particular host at their time of creation, but this assignment can be changed. In certain situations, it is even necessary. You can use the SAP HANA cockpit or the SQL Editor to execute automatic redistribution operations.

There are several occasions when tables or partitions of a table need to be moved to other servers. For example, if you plan to remove a host from your system, then you first need to move all the data on that host first to the other hosts in the system. Redistributing tables may also be useful if you suspect that the current distribution is no longer optimal.

Redistribution operations are available to support the following situations:

- You are planning to remove a host from your system.
- You have added a new host to your system.
- You want to optimize current table distribution.
- You want to optimize table partitioning.

Although it is possible to move tables and table partitions manually from one host to another, this is neither practical nor feasible for a large-scale redistribution of data.

Table Distribution

In SAP HANA cockpit, search for the Table Distribution card. From this card, select View Current Table Distribution. The Table Distribution application outlines partitioning and distribution information of tables in a distributed system. You can reduce the number of displayed tables, by using the filter function.

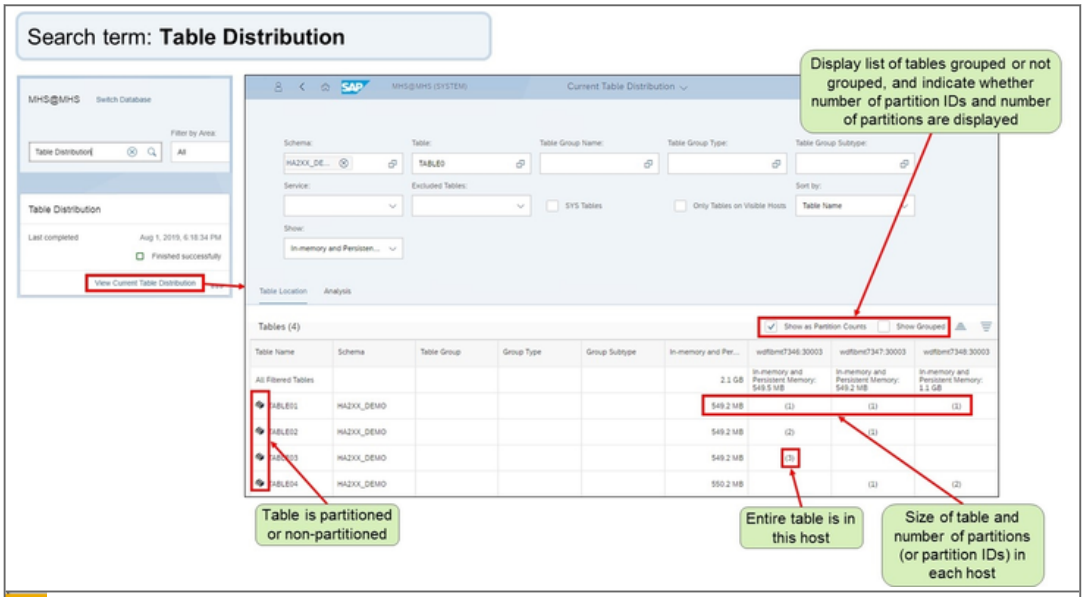


Figure 35: Table Distribution in SAP HANA Cockpit

Furthermore you can choose additional actions in the Table Distribution application.

Available operations are:

- View table distribution

Generate table redistribution plan

Save current table distribution

Restore saved tabled distribution plan

Rerun table distribution plan

Table Redistribution with SAP HANA Cockpit

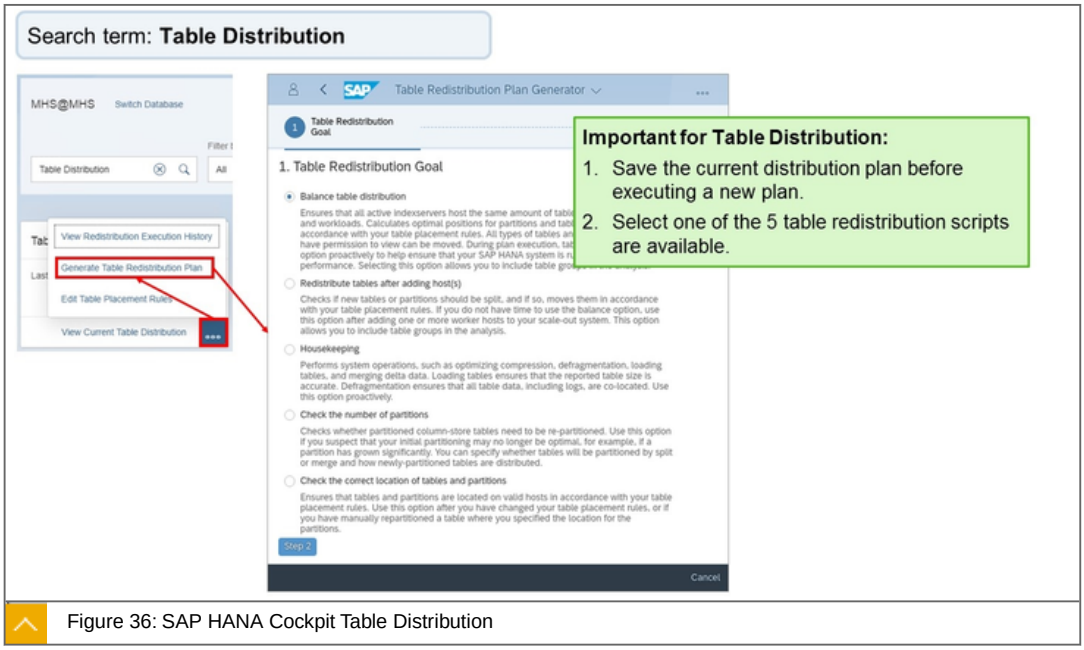


Figure 36: SAP HANA Cockpit Table Distribution

SAP HANA supports several redistribution operations that use complex algorithms, as well as configurable table placement rules and redistribution parameters, to evaluate the current distribution and determine a better distribution depending on the situation. Administrators can use the table redistribution feature in the SAP HANA cockpit to create a plan for redistributing and repartitioning tables. The administrator can review the plan and execute it.

Balance table distribution

The load on a scale-out system changes over time with the usage of the system. This option generates a plan to move tables and partitions to their proper hosts if they are currently on invalid hosts according to the rules specified in the TABLE_PLACEMENT table. The plan checks whether a split or merge is necessary and calculates optimal positions for the parts and tables. All types of tables and parts can be moved. However, only the tables that you have permission to view as catalog objects are affected.

Check the number of partitions

In a scale-out system, partitioned tables are distributed across different index servers. The location of the different partitions can be specified manually or determined by the database when the table is initially partitioned. Over time, this initial partitioning may no longer be optimal, for example, if a partition has grown significantly.

This option evaluates whether or not partitioned tables need to be repartitioned. The plan specifies how partitioned tables are repartitioned (split or merged) and how newly-created partitions are distributed. Note that this is only relevant for column-store tables. System tables, temporary tables, and row-store tables are not considered.

Redistribute tables after adding host(s)

After adding one or more worker hosts to a scale-out system, you may need to redistribute the tables across the active index servers. This option checks whether new partitions can be created and generates a plan to move the tables and table partitions as necessary.

Check the correct location of tables and partitions

This option generates a plan to move tables and partitions to their proper hosts if they are on invalid hosts according to the rules specified in the TABLE_PLACEMENT table. Only the tables that you have permission to view as catalog objects are affected.

Housekeeping

Some regular operations need to be done from time to time. This option allows you to perform various operations in the system, such as, optimize compressions, defrag, load table, and merge delta. Only the tables that you have permission to view as catalog objects are affected. Also, you must have the appropriate privileges to perform specific housekeeping operations, such as delta merge.

Table Redistribution using SQL Editor

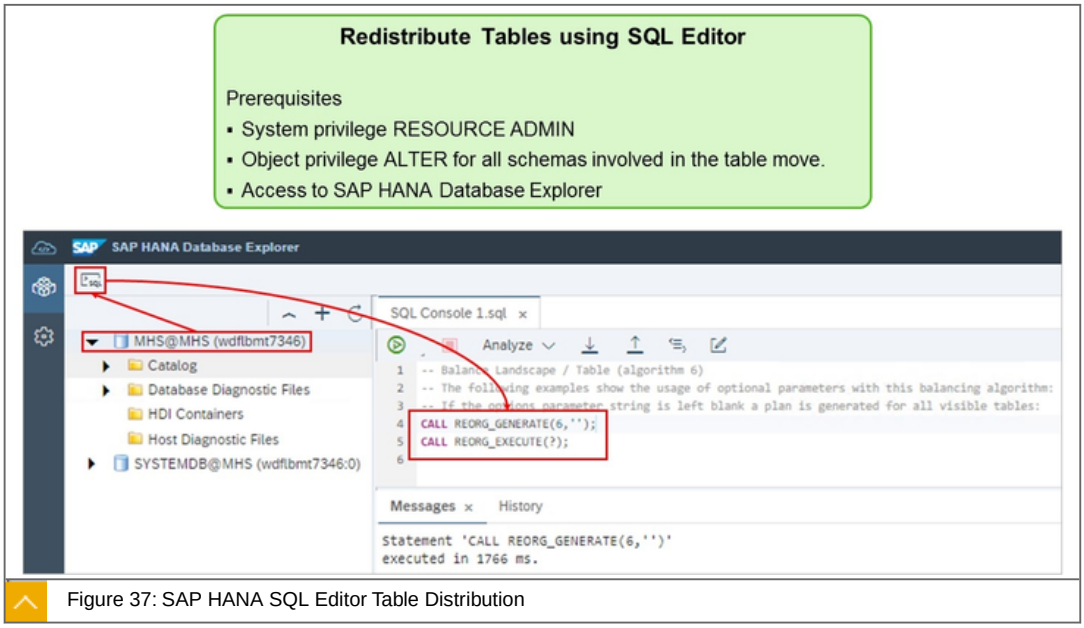


Figure 37: SAP HANA SQL Editor Table Distribution

Table redistribution can also be performed using the SQL commands. You can create an SQL script that executes all the required steps, or you can use the SQL Editor provided by the SAP HANA Database Explorer.

Table redistribution is based on the table placement rules defined in the TABLE_PLACEMENT table. These rules determine, for example, table sizes, partitioning threshold values, and preferred partition locations. Redistribution is a two-stage process: firstly, to generate the plan and secondly, to execute the plan. Separate commands are used in each stage:

1. The plan generation command is a multi-purpose tool that requires an algorithm number as a parameter to determine which actions are executed. Depending on the algorithm selected, additional optional parameter values may also be available to give more control over the execution.

2. The plan execution command takes a single parameter which is the numeric plan ID value. You can retrieve this value (REORG_ID) from the REORG_OVERVIEW system view (see System Views following).

The syntax for these commands is:

CALL REORG_GENERATE(<algorithm integer>, <optional parameter string>);

CALL REORG_EXECUTE(<plan_id>);

:

Resource admin privilege is required to call REORG_GENERATE(). The command only operates on tables and partitions that the executing user is allowed to see as catalog objects.

Generating the Plan: Algorithms and Options

The following list gives an overview of the most commonly-required algorithms.

Add server: Algorithm number: 1

Run this check after adding one or more index servers to the landscape. If new partitions can be created, a plan is generated to split the tables and move the new partitions to the newly added index servers.

Options: SCHEMA_NAME | TABLE_NAME | GROUP_NAME | GROUP_TYPE | GROUP_SUBTYPE | RECALC | NO_PLAN

Clear server: Algorithm number: 2

Moves all partitions from a named server to other servers in the landscape.

Options: USE_GROUP_ADVISOR

Save Algorithm number: 4

Save the current landscape setup.

Restore Algorithm number: 5

Restore a saved landscape setup. Enter the plan ID value as the optional parameter value.

Balance landscape: Algorithm number: 6

This function checks if tables in the landscape are placed on invalid severs according to the table placement rules, and checks if a split or merge is necessary to achieve optimal positions for the partitions and tables, and to evenly distribute tables across the index server hosts.

Options: SCHEMA_NAME | TABLE_NAME | GROUP_NAME | GROUP_TYPE | GROUP_SUBTYPE | RECALC | NO_PLAN | NO_SPLIT | SCOPE

Check number of partitions Algorithm number: 7

This function checks if partitioned tables need to be repartitioned and creates a plan to split tables if the partitions exceed a configured row count threshold. No optional parameter.

Execute Group Advisor Algorithm number: 12

Calls the Group Advisor and creates an executable plan from its output.

The Group Advisor identifies tables which are often used together so that during redistribution they can be located together on the same node to avoid cross-node communication in the landscape.

Check table placement Algorithm number: 14

Check current landscape against table placement rules and (if necessary) provide a plan to move tables and partitions to the correct hosts.

Options: LEAVE_UNCHANGED_UNTOUCHED | KEEP_VALID | NO_SPLIT

Rerun plan Algorithm number: 15

Rerun failed items from previously executed plans.

Option: RERUN_ALL

HousekeepingAlgorithm number: 16

Perform housekeeping tasks. Additional privileges may be required for specific actions.

Options: OPTIMIZE_COMPRESSION | DEFRAG | LOAD_TABLE | MERGE_DELTA | ALLOptional

Pre-defined Table Placement Scenarios

For specific applications, SAP provides recommendations regarding partitioning and table distribution configurations.

SAP BW powered by SAP HANA

All required steps and recommended settings for SAP BW on HANA 2 are described in SAP Note: 1908075 — “BW on SAP HANA: Table placement and landscape redistribution”. This includes a zip file with documentation and SQL code to configure various scenarios covering a range of TABLE_PLACEMENT settings depending on the node size (TB per node) and the number of master and slave nodes.

SAP Business Suite Powered by SAP HANA

SAP Note: 1899817 — “SAP Business Suite on SAP HANA database: Table Placement” includes configuration scripts to set up partitioning and distribution for Suite and S/4HANA for various support package stack releases.

SAP S/4HANA

Starting with SAP S/4HANA 1610 FPS1, scale-out is supported and can be applied in special scenarios. Application data tables are grouped together according to application area and can be placed as a table group on a specific server. In this way, it is possible to use scale-out in SAP S/4HANA.

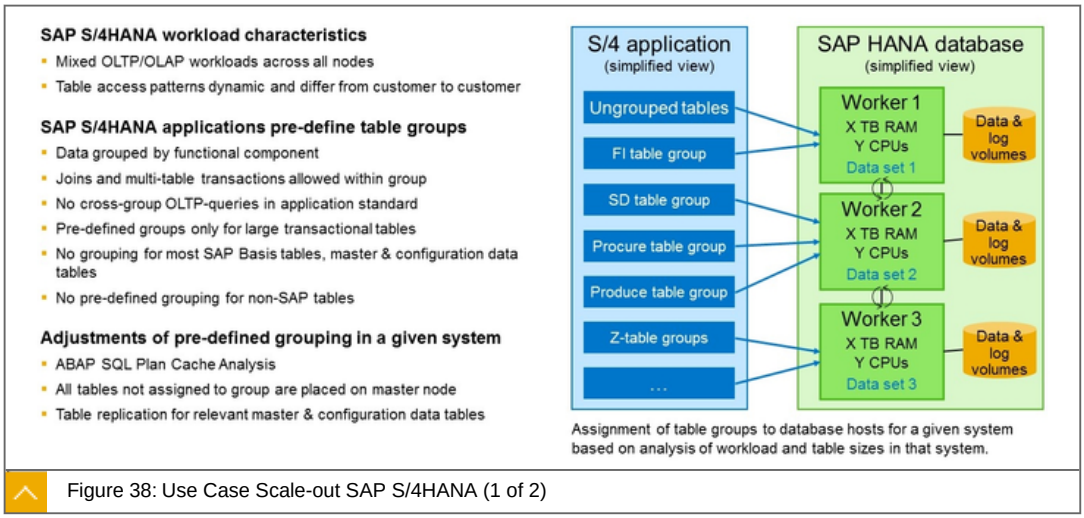


Figure 38: Use Case Scale-out SAP S/4HANA (1 of 2)

For details of scale-out options for SAP S/4HANA, refer to SAP Note: 2408419 — “SAP S/4HANA - Multi-Node Support”. This note includes scripts and configuration settings, as well as detailed documentation about table groups and migration.

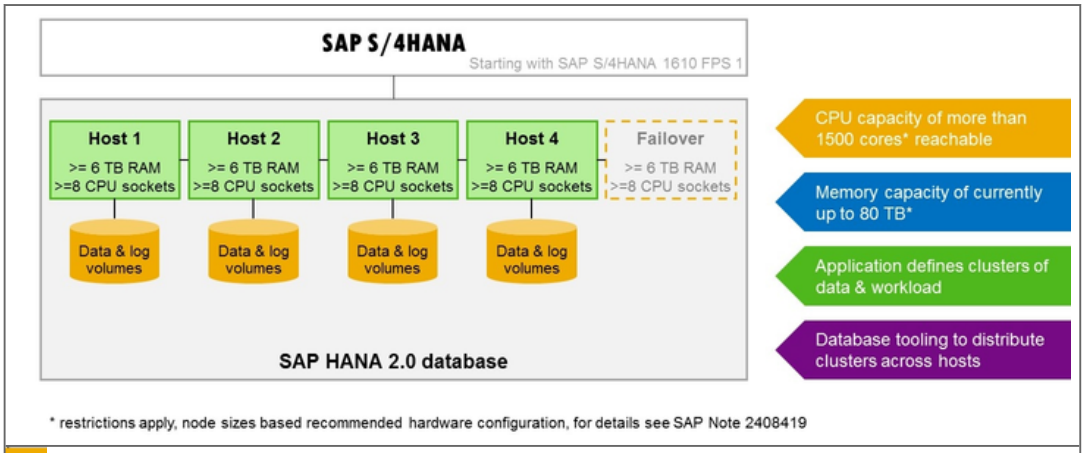


Figure 39: Use Case Scale-out SAP S/4HANA (2 of 2)

With the mixed workload on an S/4HANA system, it still makes sense to first scale-up as far as possible, before going into scale-out. So if the first node is bigger than 6 TB and 8 CPU sockets, then scale-out is allowed. See the previous figure for an example.

SAP BW/4 HANA

All the required steps and recommended settings for SAP BW/4 on HANA 2 are described in SAP Note: 2334091 — “BW/4HANA: Table Placement and Landscape Redistribution”. This includes a zip file with documentation and SQL code to configure various scenarios covering a range of TABLE_PLACEMENT settings depending on the node size (TB per node) and the number of master and slave nodes.

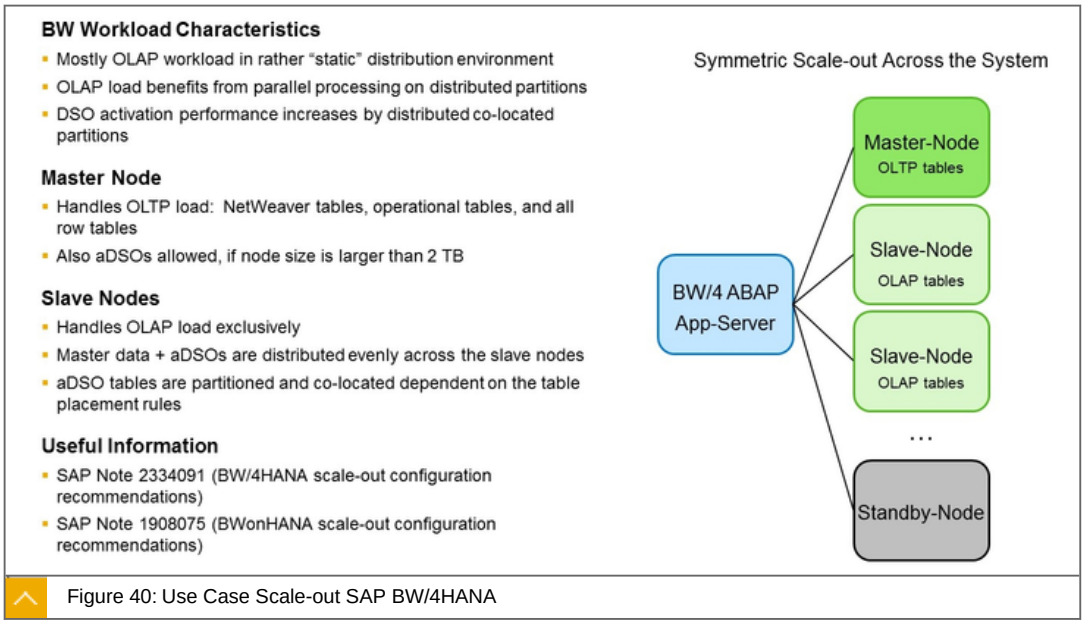


Figure 40: Use Case Scale-out SAP BW/4HANA

SAP provides recommendations regarding table distribution configurations. For these scenarios, SQL implementation scripts and detailed documentation is provided in SAP Notes.

Pre-defined Table Placement Scenarios	SAP Note
SAP BW powered by SAP HANA	1908075
SAP Business Suite Powered by SAP HANA and S/4HANA	1899817
SAP S/4HANA	2408419
SAP BW/4HANA	2334091
Enable BPC HANA table distribution	2003863

Figure 41: Pre-defined Table Placement Scenarios

LESSON SUMMARY
You should now be able to:

Perform Table Placement Tasks

Unit 2
Lesson 5

Reconfiguring a Scale-Out SAP HANA System

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Reconfigure a scale-out SAP HANA system

Reconfiguration of a Scale-Out SAP HANA System

Business Example

As an SAP HANA database administrator, you must be able to reconfigure your company's high availability scale-out SAP HANA systems. You require hands-on experience with reconfiguring multi-host SAP HANA systems.

Reconfigure a Scale-out SAP HANA System

The SAP HANA database supports high availability in a distributed system by providing for host auto-failover. If an active host fails, for example, because of a hardware failure, standby hosts can take over and ensure the continued availability of the database.

In SAP HANA cockpit 2.0, you can monitor the status of individual hosts in the Host Failover application. To start the Host Failover application, open SAP HANA cockpit 2.0 and navigate to the Aggregate Health Monitor application. In the Aggregate Health Monitor, select the SYSTEMDB@<SID>link to open the SAP HANA Cockpit Overview screen.

On the SAP HANA Cockpit Overview screen, search for the Configure Host Failover application, and select the Configure host failover link. The same application is also available from the tenant database, <SID>@<SID>, but this task is normally something you perform from SYSTEMDB.

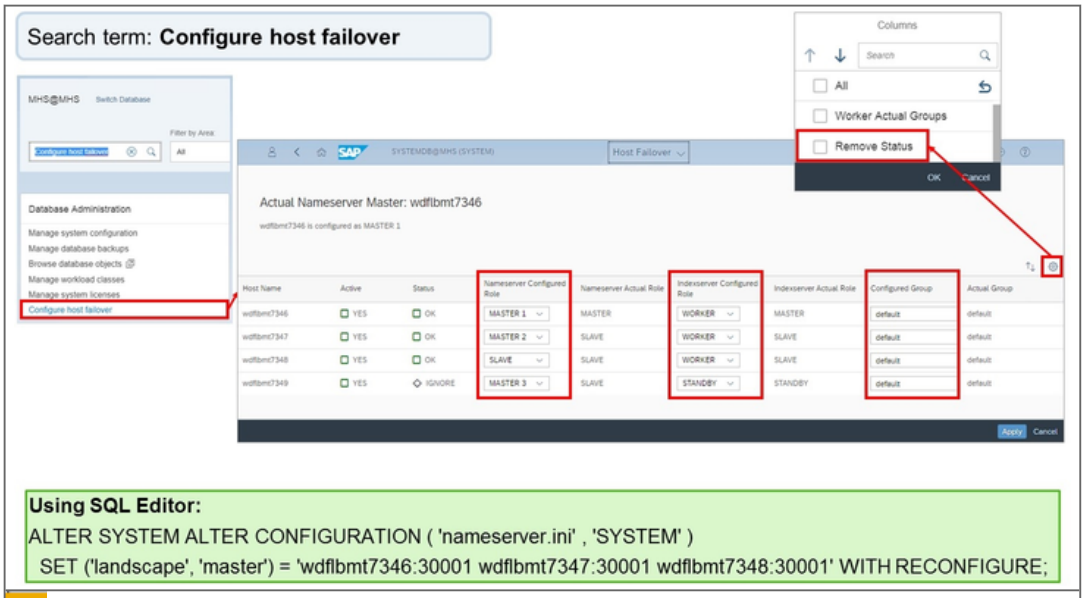


Figure 42: SAP HANA Cockpit Host Failover

Host roles for failover are normally configured during installation. Using the SAP HANA cockpit, you can monitor the status of individual hosts and switch the configured roles of hosts. You cannot increase or decrease the number of worker hosts and standby hosts with respect to each other.

The primary reason for changing the configured roles is to prepare for the removal of a host. In this case, change the configured role of the name server host to SLAVE and the configured role of the index server host to STANDBY before stopping the database instance on the host and removing the host.

The reconfiguration is done by setting the master parameter in the landscape section of the nameserver.ini file.

With the following SQL command, you specify which three hosts are master candidates:

```
ALTER SYSTEM ALTER CONFIGURATION ( 'nameserver.ini' , 'SYSTEM' )
  SET ('landscape', 'master') = 'wdflbmt7346:30001 wdflbmt7347:30001
wdflbmt7348:30001' WITH RECONFIGURE;
```

With the following SQL command, you specify which host is the master:

```
ALTER SYSTEM ALTER CONFIGURATION ( 'nameserver.ini' , 'SYSTEM' )
  SET ('landscape', 'active_master') = 'wdflbmt7347:30001' WITH
RECONFIGURE;
```

In the SAP HANA Cockpit - Host Failover, you can add or remove columns in the display by choosing the gear button in the top-right corner. When adding or removing hosts from a multi-host SAP HANA system, the remove status column indicates the status of the table redistribution operation used to move data off the index server of a host that you plan to remove.

Before you can remove an active host from a single-container system, you must move the tables on the index server of this host to the index servers on the remaining hosts in the system. Once the value in the removal status column changes to REORG FINISHED or REORG NOT REQUIRED, you can physically remove the host using the SAP HANA lifecycle management tool, hdblcm(gui).

If your system is configured as a multiple-container system, you must remove tenant-specific services first and then remove the host using the SAP HANA database lifecycle manager.

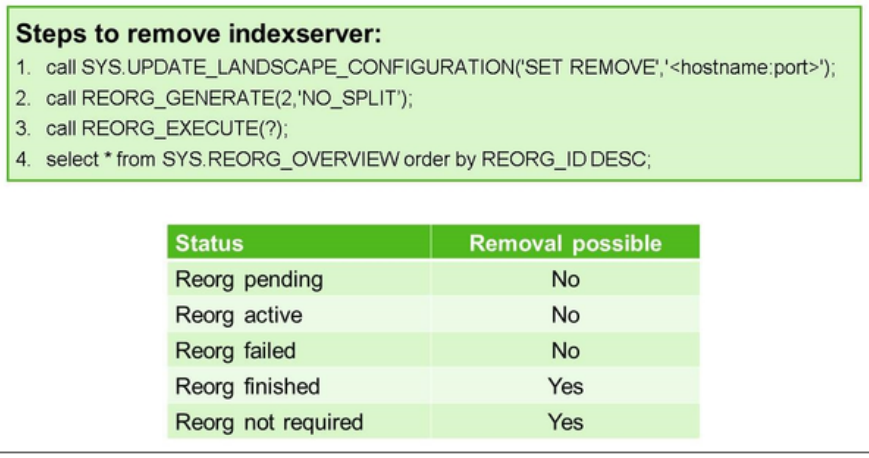


Figure 43: Host Removal Status

The following statuses are possible:

<Empty>: The host has not been marked for removal.

REORG PENDING: A redistribution operation is required to move tables to other hosts.

REORG ACTIVE: A redistribution operation is in progress. For more information, you can query the system tables SYS.REORG_OVERVIEW and SYS.REORG_STEPS.

REORG FAILED: A redistribution operation was executed and failed. For more information, query the system table SYS.REORG_STEPS.

REORG FINISHED: A redistribution operation has completed. The host can be uninstalled.

REORG NOT REQUIRED: A redistribution operation is not required. The host can be uninstalled.

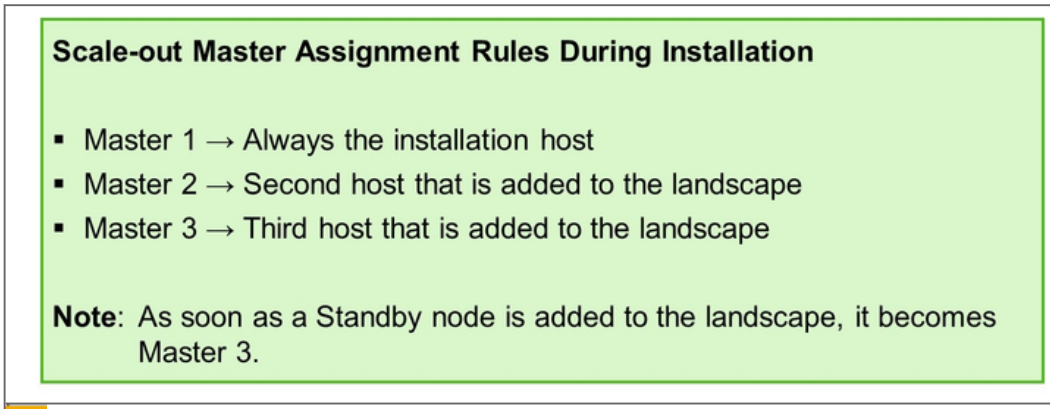


Figure 44: Installation Scale-out Master Assignment Rules

During the SAP HANA installation of a multi-host system, the optimal auto-failover configuration is set up.

The optimal configuration is described in the following setup:

Master 1: Assigned to the node where the installation is performed.

Master 2: Assigned to the first additional node that is assigned to the multi-host system.

Master 3: Assigned to the first standby node assigned to the multi-host system. If there is no standby host configured, the second additional node that is assigned to the multi-host system is used. A setup without at least one standby node will not be a useful high-availability scenario.

In our training class environment, the SAP HANA installation that we performed in the exercise “Install a High Availability SAP HANA System” has set up the optimal auto-failover configuration, as shown in the previous figure. For educational purposes, the next exercise “Reconfigure a Scale-out SAP HANA System” shows you how to reconfigure the auto-failover configuration using the Host Failover application in SAP HANA cockpit 2.0.

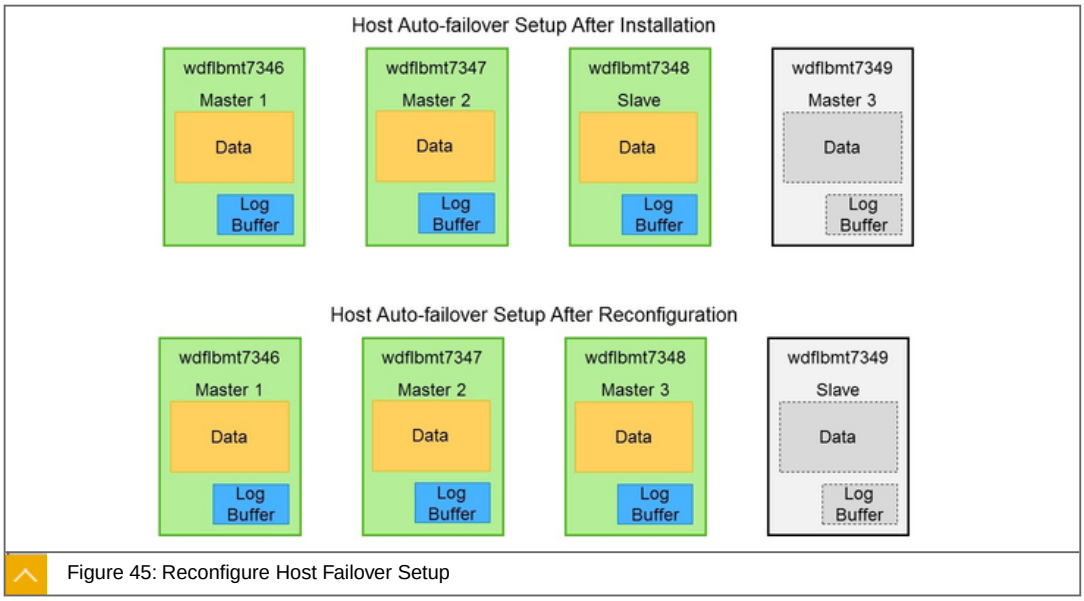


Figure 45: Reconfigure Host Failover Setup

In the next three exercises, the failover behavior is shown. This changed configuration behavior clearly shows us what happens when a slave node or the master 1 node fails. In the last exercise, we return to the optimal auto-failover configuration and demonstrate what happens when the master 1 node fails, but the master 3 node is assigned to the standby server.

Host Auto-failover Required Authorizations

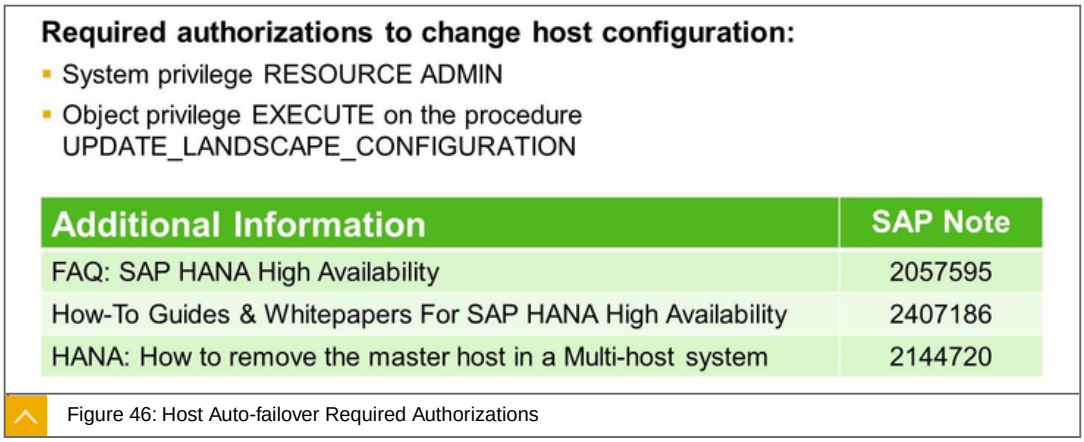


Figure 46: Host Auto-failover Required Authorizations

LESSON SUMMARY
You should now be able to:

- Reconfigure a scale-out SAP HANA system

Unit 2
Lesson 6

Understanding Failure of an SAP HANA Slave Node

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand what happens during a failure of a slave node

Failure of an SAP HANA Slave Node

Business Example

As an SAP HANA database administrator, you need to understand the SAP HANA host auto-failover concept. To better understand this feature, you need to have hands-on experience with a slave node failing in a multi-host SAP HANA system.

Failure of an SAP HANA Node

Host auto-failover is a local fault recovery solution that can be used in addition to, or as an alternative measure to, system replication. One (or more) standby hosts are added to an SAP HANA system and configured to work in standby mode. If they are in standby mode, the databases on these hosts do not contain any data and do not accept requests or queries. This means that they cannot be used for other purposes, such as quality or test systems.

When a primary (worker) host fails, a standby host automatically takes its place. If neither the name server process hdbnameserver, nor hdbdaemon respond to network requests (because the instance is stopped or the OS has been shut down or powered off), a host is marked as inactive and an auto-failover is triggered. Since the standby host may take over operation from any of the primary hosts, it needs shared access to all the database volumes. This can be accomplished by a shared, networked storage server, by using a distributed file system, or with vendor-specific solutions that use an SAP HANA programmatic interface, the Storage Connector API, to dynamically detach and attach (mount) networked storage upon failover.

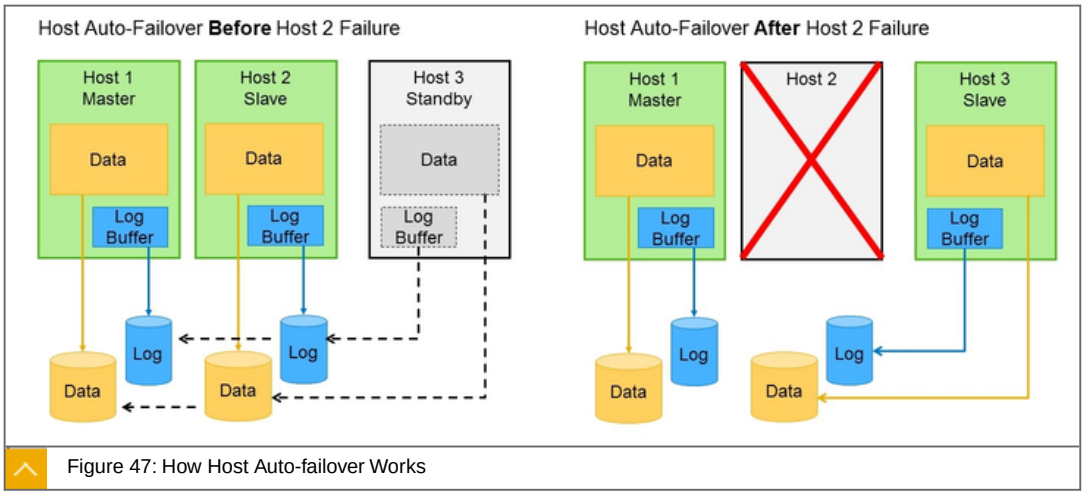


Figure 47: How Host Auto-failover Works

As implied by the name, the host auto-failover capability of SAP HANA is characterized as follows:

Failover is performed at the host level. All services of a host are moved to another host. The failure of a single process (service) does not trigger a failover.

The failover happens automatically as an integral feature of SAP HANA. No external cluster manager is required.

Data consistency is a key requirement. Data might be corrupted if a failed host (like the original host 2 in the previous figure) is allowed to restart and write data to disk in parallel to the failover host (the new slave host 3 in the previous figure).

To ensure data consistency at all times, it must be guaranteed that a failover does not happen (or at least does not succeed and may not cause corrupt data) if the failed host can potentially still write data. To achieve this, the SAP HANA host auto-failover uses a combination of heartbeat and fencing.

Heartbeat

The following types of heartbeat are used to check if another host is active as the master before starting the current host as the master or performing a failover:

TCP communication-based heartbeats:

- Ping from nameserver to nameserver with SAP HANA internal communication protocol
- Ping from nameserver to hdbdaemon with SAP HANA internal communication protocol

Storage-based heartbeats:

The current master nameserver periodically updates heartbeat files located on different storage partitions:

- Shared storage for the SAP HANA binaries
- Storage partition 1 for the master node’s data

These types of storage are typically connected with networks other than the inter-node network used for service-to-service communication (such as fiber channel for SAN or dedicated Ethernet for NFS) and therefore these heartbeats provide additional value.

Fencing

In rare cases, the heartbeats cannot detect if another host is alive, for example in split-brain situations where no communication is possible between hosts. I/O fencing ensures that the other side does not access the data or log storage any more.

The SAP HANA Storage Connector API, together with a specific Storage Connector, allows usage of different types of storage and network architecture to ensure proper I/O fencing:

SAN storage: the SAP HANA Fiber Channel Storage Connector [2] using SCSI-3 persistent reservations (SCSI-3 PGR).

NFSv3: used without file locking, but with a Storage Connector provided by certified storage vendors. This type of Storage Connector implements a Shoot The Other Node In The Head (STONITH) call to reboot a failed host.

If an NFSv3 client dies (that is, the SAP HANA server), the file locks are not released on the NFS server side resulting in a deadlock for any host that wants to access these files. Using the nolock mount option solves the locking problem, but with this option, data is not protected against parallel reading and writing from different hosts. To solve this, STONITH must be implemented.

NFSv4 or cluster file systems like GPFS: using file locks. A Storage Connector is not required here as these file locks reliably prevent false access. However, a STONITH type Storage Connector is provided by some storage vendors to speed up failover.

Review the SAP HANA Multi-Host Configuration from the Command Line

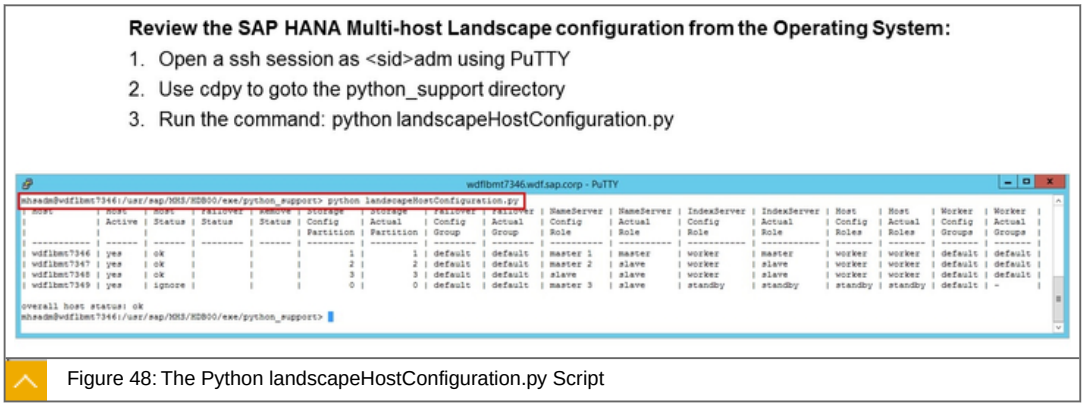


Figure 48: The Python landscapeHostConfiguration.py Script

The SAP HANA multi-host configuration can also be viewed at the operating system level. There is a Python script called landscapeHostConfiguration.py in the $DIR_INSTANCE/exe/python_support folder. Running the script as shown in the previous figure provides an overview of the configuration.

The following host columns are shown in addition to the SAP HANA cockpit 2.0 view by this script:

STORAGE_CONFIG_PARTITION / Storage Partition (Configured - new in SPS 12): The stable sub-path to reassign the same storage partition after failovers.

WORKER_CONFIG_GROUPS / Worker Groups (Configured – new in HANA 2 SPS 00): The stable classification values to assign hosts to logical worker groups.

WORKER_ACTUAL_GROUPS / Worker Groups (Actual – new in HANA 2 SPS 00): The current classification values to assign hosts to logical worker groups.

The return code may be consumed by cluster managers (for example, for SAP HANA system replication) to come to a decision on the system health state, as follows:

0 = Fatal. For example, database offline.

1 = Error. For example, a failover did not happen, because there was no standby host available.

2 = Warning. For example, a failover is possible.

4 = OK

5 = Ignore. For example, the system has switched roles (failover), but is fully functional.

A return code >= 4 indicates normal system operation. When the system is stopped, this script can also be used, but fills only a subset of the columns.

Host Failure Detection

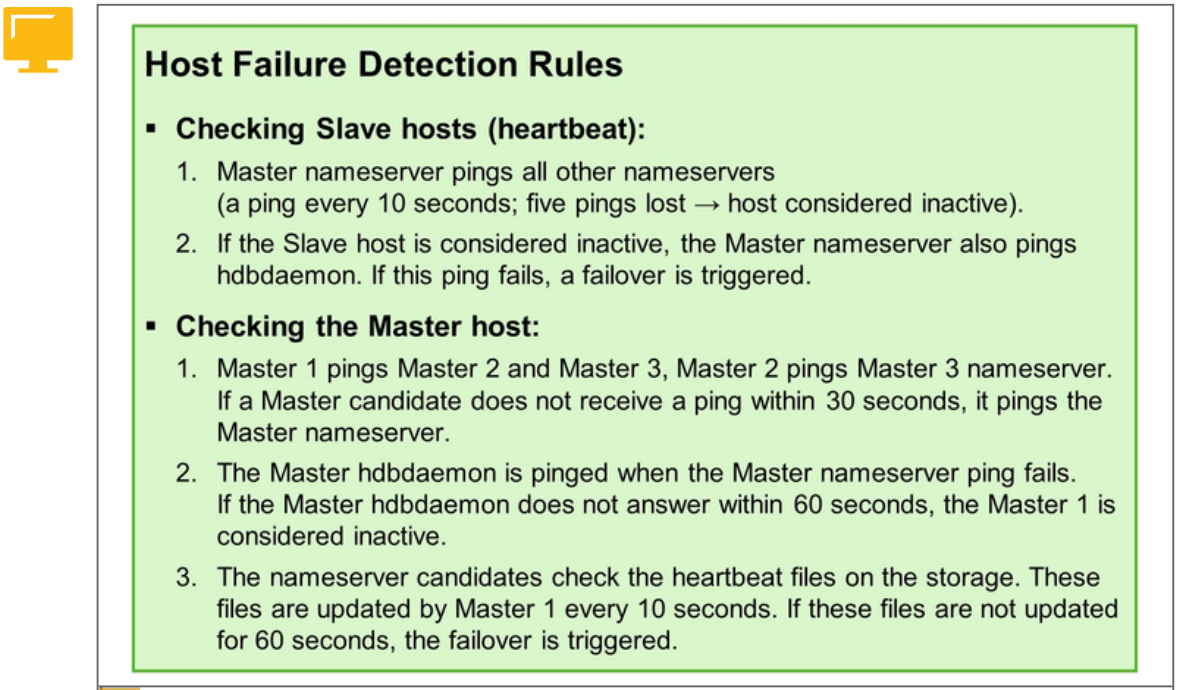


Figure 49: Host Failure Detection Rules

A host failure is any dysfunctional state of a host that affects the communication between the hosts of a distributed SAP HANA system. To check the functional state of a host, the name servers regularly send a ping on the internal network communication layer to name servers on other hosts. An additional ping to the hdbdaemon process is executed in the case where the remote name server does not reply repeatedly. Only when both services do not reply in time, is the host considered to have failed.

A crash of a single service does not trigger failover, because services are normally restarted by the hdbdaemon. If a service is not able to restart for any reason, it is assumed that it is not be able to start on another host either.

An exception is if the name server aborts itself during startup if the storage connector returns an error. It then instructs hdbdaemon to shut down the whole database instance on the host including the hdbdaemon itself, which allows failure detection and failover processing by other hosts.

Checking Slave Hosts

The name server communication heartbeat: The current master name server pings all other name servers every 10 seconds. If a name server was active and five pings have failed (either immediately or after a 60 second ping timeout), the name server is considered inactive. By pinging multiple times, SAP HANA can recover from short network outages without triggering a failover.

The hdbdaemon communication heartbeat: If a slave name server was considered inactive (or had set itself to inactive), the master name server pings the slave hdbdaemon process. If the hdbdaemon ping fails (either immediately or after a 60 second ping timeout), the host is considered as inactive and a failover is initiated.

Checking the Master Host

The name server communication heartbeat: Name server candidates, which are not currently the master, ping other candidates with lower priority every 10 seconds. Together with the slave name server heartbeat described earlier (current master name server pings all other name servers), normally MASTER1 pings MASTER2 and MASTER3, and MASTER2 pings MASTER3. If a master candidate does not receive any ping within 30 seconds, it pings the master name server itself.

The hdbdaemon communication heartbeat: If the ping to the master name server fails, the hdbdaemon process on the master host is pinged. If the hdbdaemon does not answer within 60 seconds, the current master host is considered inactive.

The name server storage heartbeat: The name server candidate host checks the heartbeat files for changes for a period of 60 seconds. Those files are updated by the current master name server every 10 seconds with the hostname and a random string. A failover begins only if all files do not show any sign of changes for 60 seconds.

Slave Host Failover to a Standby Host

When a failure is detected and a replacement host is determined, the actual failover process starts.

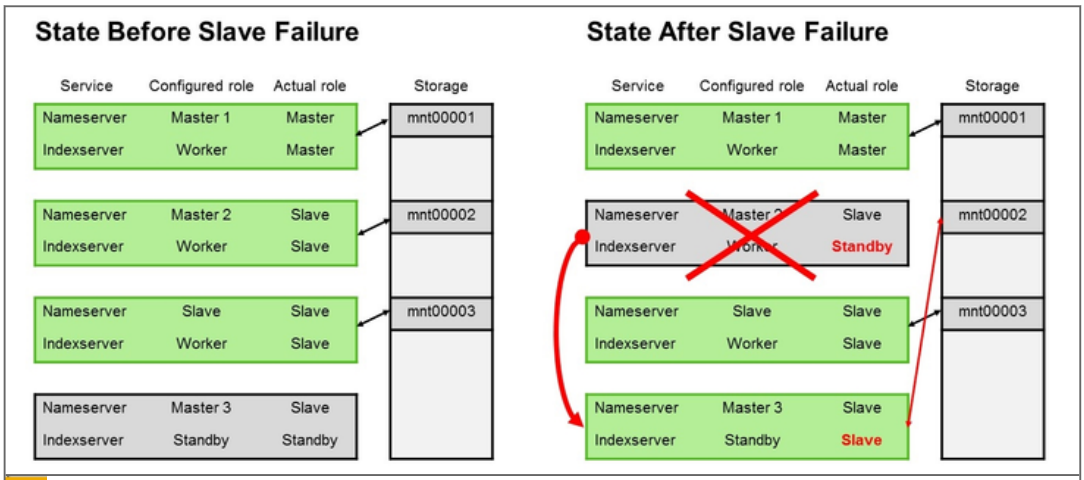


Figure 50: Host Auto-failover Slave Failure

The previous figure is a visualization of a slave host failover to a standby host. On the left, the original state of the system is shown. On the right, the second host fails and its role is moved to the fourth host.

Failover step-by-step:

1. Target host selection

   If there is a standby host with an exact match of corresponding actual host roles, it is used.

   If there is a standby host with one of the roles that corresponds to the failing host, it is used.

   If the failing host has an SAP HANA worker role, any unassigned standby is used.

2. The master name server calls the stonith() method of all installed HA/DR provider hooks and the Storage Connector stonith() method. Typically the stonith() method is only implemented in NFSv3-related storage connectors and reboots the failed host.

Note:
If STONITH fails, failover is aborted and all hosts remain in their old roles.

3. Swap actual services, host roles, storage partition number, and volume IDs of all services between both hosts in the topology and inform all other hosts.
4. The master name server (which selected a replacement host), calls the name server on the target host to perform the failover.
5. The host that was promoted to a new role calls the Storage Connectors attach() method to acquire the correct storage partition (if applicable) and calls the failover() method of all installed HA/DR provider hooks.

Note:
If this fails, the host stops. If there are still standby hosts available, another failover is triggered and this host is set to ERROR.

6. Reconfigure running standby services to load their newly assigned volume.

Note:
If this fails, this is like a service failure and does not initiate a further failover.

7. Reconfigure hdbdaemon to start/stop services that should run on only one of the two hosts.

Note:
If this fails, this is like a service failure and does not initiated a further failover.

The master name server is the only entity in the whole system that is able to make a failover target host selection. Since the master has mechanisms to avoid split brain situations, there is conceptually no split brain situation possible for slave hosts. If a slave loses its connection to the master name server, it waits and is notified by the new master. If a slave cannot connect to a master during startup, it terminates itself.

Exercise “Failing SAP HANA Slave Node” Explained

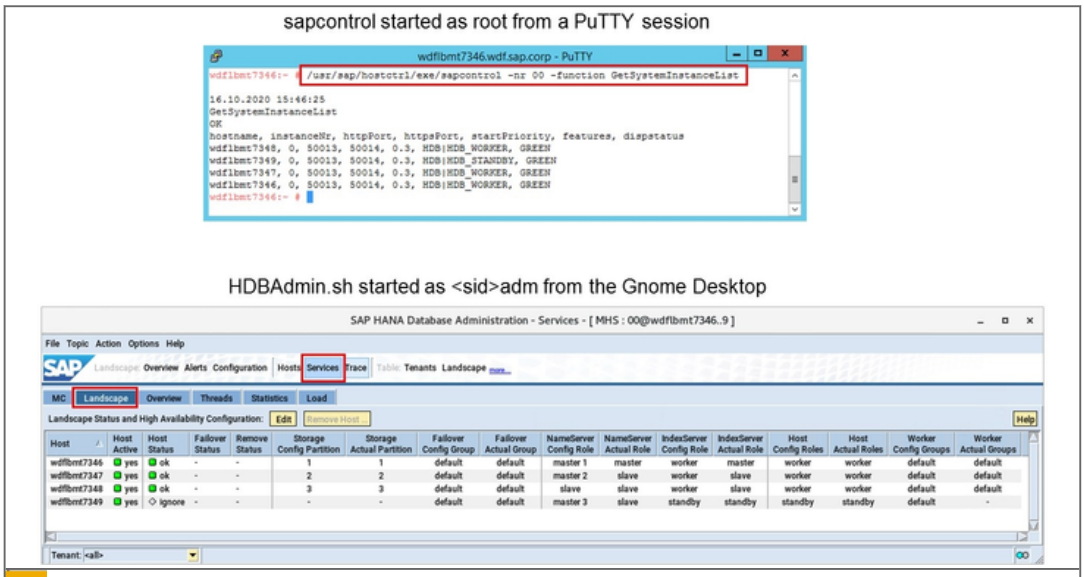


Figure 51: Monitor the Host Auto-failover

In this exercise, all admins monitor their SAP HANA node with the tools HDBAdmin and sapcontrol. HDBAdmin is a graphical SAP Support tool that can be used to monitor the SAP HANA system in real time. With sapcontrol, the SAP HANA admin can request the SAP HANA instance list from the command line.

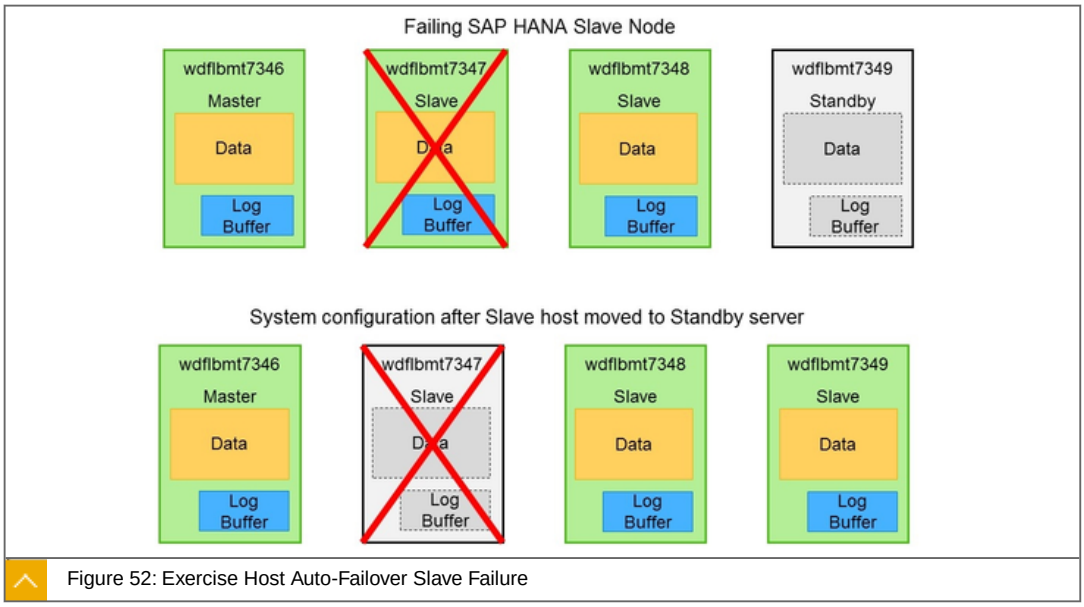


Figure 52: Exercise Host Auto-Failover Slave Failure

The admin on wdflbmt7347 simulates a slave node failure, by executing an <sid>adm a HDB kill command. Due to this failure, the auto host-failover steps can be monitored in the HDBadmin tool and by using sapcontrol at the command line.

LESSON SUMMARY
You should now be able to:

- Understand what happens during a failure of a slave node

Unit 2
Lesson 7

Understanding Failure of the SAP HANA Master Node

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand what happens during a failure of the master node

Failure of the SAP HANA Master Node

Business Example

As an SAP HANA database administrator, you need to understand the SAP HANA host auto-failover concept. To better understand this feature, you need to have hands-on experience of a master node failing in a multi-host SAP HANA system.

Failover Algorithm Explained

In contrast to other high availability solutions, SAP HANA does not use a quorum consisting of multiple SAP HANA hosts to decide which host can become master at initial startup or master failover. With heartbeats and fencing, a single host can reliably decide initial startup or master failover.

Master Host Failover with Standby Host but All Master Candidates in Use (Double Failover)

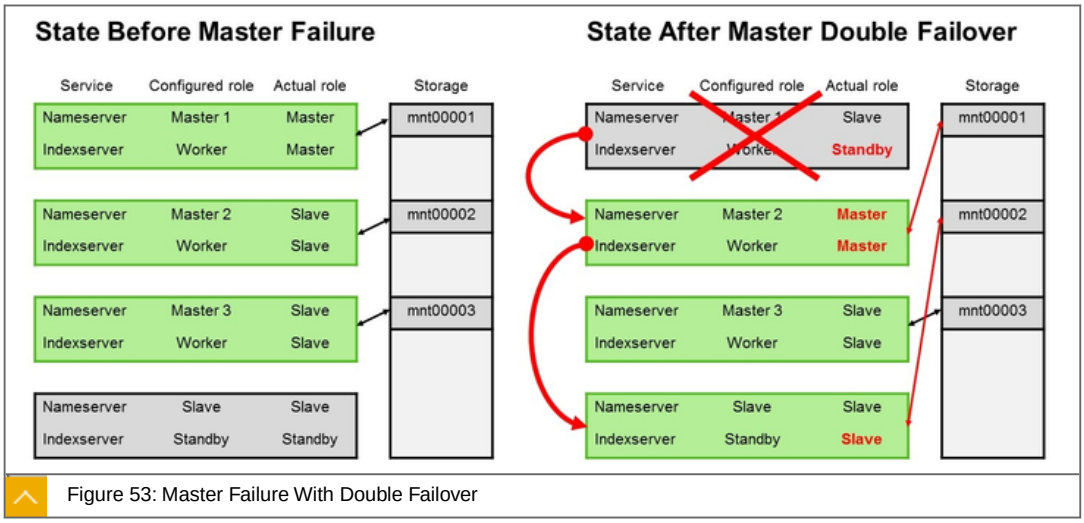


Figure 53: Master Failure With Double Failover

The previous figure shows a master host failover to a slave host. On the left, the original state of the system is shown. On the right, the first host fails and its master role is moved to the second host. The original slave role of the second host is failed over to the standby host.

Failover step-by-step:

1. If no master name server candidate with standby as an actual index server role is available, one of the master candidates currently used as index server slave is chosen as the new master.
2. The failover steps for the master host are like the scenario described in the figure, Master Host Failover Without Available Standby Hosts.
3. The previously assigned slave is marked as failed and enters the failover queue. Because a standby host is available, slave failover starts shortly after master failover.
4. Both failovers are executed in parallel.

Master Host Failover to a Standby Host

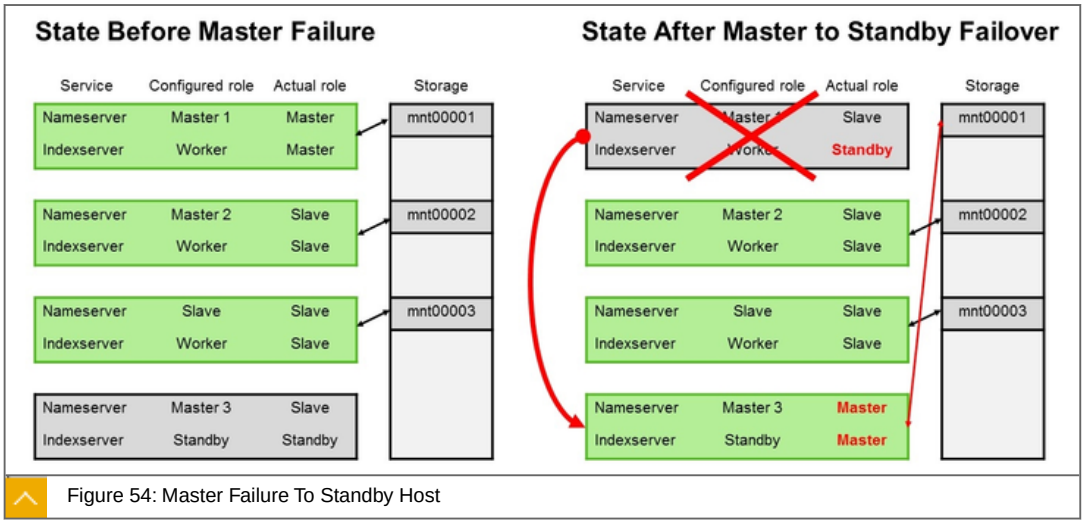


Figure 54: Master Failure To Standby Host

The previous figure shows a master host failover to a standby host. On the left, the original state of the system is shown. On the right, the first host fails and its role is moved to the fourth host.

Failover step-by-step:

1. The name server master candidate with the highest priority (= smallest number in the configured name server role) detects the failure condition and initiates the failover.
2. If a name server candidate is available that is currently a standby host, the failover is forwarded to this host. This avoids a double failover (see the second example following).
3. The failover includes the same steps as in the slave host failover scenario described earlier.
4. The name server reloads its persistence from disk.

Target Host Selection

This section describes the selection process of the replacement host. Beginning with SPS11, the actual host roles (HOST_ACTUAL_ROLES) are considered.

For SAP HANA 1.0 SPS11 and newer:

If there is a standby host with an exact match of corresponding actual host roles, it is used.

If there is a standby host with one of the roles that corresponds to the failing host, it is used.

If the failing host has an SAP HANA worker role, any unassigned standby is used.

For SAP HANA 1.0 SPS10 and older:

If there is a standby host, it is used.

If there are multiple equivalent options available, the first host is used.

The search steps are restricted to the same failover group, unless global.ini/[failover]/ cross_failover_groups=false was configured.

If no host is available, no failover happens and HOST_STATUS shows ERROR.

Master Host Failover Without Available Standby Hosts

Distributed landscapes without standby hosts may also perform a failover to ensure that the master host is always available. Of course, a slave host (and all tables located there) is inaccessible after failover.

This failover mechanism can be disabled by removing the name server roles MASTER 2 and MASTER 3 in the SAP HANA cockpit. Disabling is required if you use (not recommended) local storage on each host or the landscape is controlled by an external cluster manager.

The wait timeout of a name server slave (non-master candidate) on a system restart is different than that of a master candidate. The number of retries to reach the master name server before aborting the startup is controlled with the following parameter: nameserver.ini/[failover]/slave_to_master_startup_retries=10 .

Because the wait interval after one unsuccessful retry is 5 seconds, the default parameter value of 10 leads to a maximum wait time of 50 seconds.

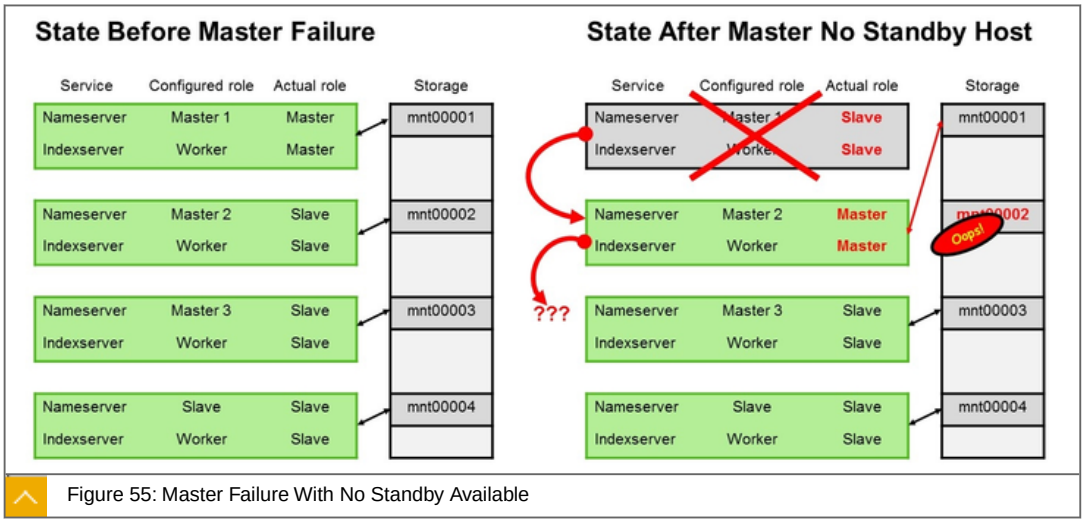


Figure 55: Master Failure With No Standby Available

The previous figure shows a master host failover to a slave host. On the left, the original state of the system is shown. On the right, the first host fails and its role is moved to the second host. The original role of the second host is not available until a standby host is added to the system or the failed first host is re-activated.

Failover step-by-step:

1. The name server master candidate with the highest priority detects the failure conditions and executes the failover steps itself.
2. The new master name server calls the stonith() method of all installed HA/DR provider hooks and the Storage Connector stonith() method (if applicable) to reboot the failed host.
   a. If STONITH fails, the failover is aborted and the new master shuts itself down.
   b. The (possibly) remaining third master candidate then retries the failover.
   c. If this also fails, no master is available throughout the whole landscape and the slave hosts eventually shut themselves down.
3. The new master stops all its services (except hdbdaemon and nameserver).
4. The new master calls the Storage Connector's detach() method for the old storage partition, the attach() method for the storage partition 1 (mnt00001 directory) and calls the failover() method of all installed failover hooks:
   a. If this fails, failover is aborted, and the new master shuts itself down.
   b. The (possibly) remaining third master candidate then retries the failover.
   c. If this also fails, no master is available throughout the whole landscape, and the slave hosts shut themselves down.
5. The new master name server loads its persistence from disk.
6. Currently existing services, host roles, storage partition number, volume IDs of all services are swaped between both hosts in the topology and all name servers are informed.
7. The hdbdaemon process is reconfigured, which starts all the required services.
8. The role of the displaced slave host remains inactive; the system is only partially available.

Host Auto-Failover vs External Cluster Manager

Instead of using the built-in SAP HANA host auto-failover, you could monitor and (re)start virtualized hosts on different hardware with an external cluster manager. With multiple SAP HANA instances, this would have the advantage that fewer standby hosts would be needed, but on the other hand, all failure detection and fencing logic would have to be implemented externally. To avoid unnecessary SAP HANA-controlled master failovers, the name server MASTER 2 and MASTER 3 roles can be removed as described previously.

Automatic Host Shutdown by Service Failures

For every service, a fixed number of restarts can be defined after which the daemon stops itself. The relevant parameters are set for each service type in daemon.ini:

```
# If set to true the daemon will shut down all services on the host if
# this service cannot start
startup_error_shutdown_instance=true
# Number of retries if a service fails in startup procedure
startup_error_restart_retries=4
```

The name server is the only service that has the latter parameter set to true by default. This means that any problem involving a constant name server crash, stops the daemon eventually. For instance, the presented settings may be used for the index server if recurring start-up problems of that service should stop the affected database instance.

SAP HANA and Split Brain

In SAP HANA's master/slave/standby failover solution, there is only one entity in the whole system that can make failover decisions, that is, the master name server. A slave or standby host never executes a failover by itself. Therefore, only the master host must be considered for split brain situations.

SAP HANA would run into a split-brain situation if multiple hosts try to become master name server/index server and access the same set of data (persistence) from disk. This would irreparably destroy the data. To overcome this problem, SAP HANA uses I/O fencing to prevent the other host from accessing the storage, as follows:

- SAN storage: The storage devices are locked by the current active host with SCSI-3 persistent reservations. If another host tries to mount those devices, the old host automatically loses write permissions and the services abort themselves.
- NFSv3 shared storage: The NFSv3 file lock implementation cannot be used as locks would not be released if an NFSv3 client dies, so a STONITH procedure must be provided by the storage vendor, which reboots a failed host.
- NFSv4 shared storage or cluster file systems like GPFS: The file locking implementation works reliably across hosts. Non-availability of a host, and thus lock release, is handled by the file system. A host that tries to open a persistence that is already open fails and aborts itself.

Communication network and storage network based heartbeats are used to detect activeness of other hosts and prevent unnecessary failover attempts. If the target master host detects that another master is still active, it terminates itself to let the other master continue. Without this, different hosts could try to become master and would fence each other repeatedly.

In a split brain situation, a quorum is sometimes used to decide, which side should 'survive'. This makes sense in stateless compute clusters to have the bigger parts of resources remaining active. However, in SAP HANA, tables are bound to specific storage partitions and service instances. Tables in the other partition would not be accessible and applications typically cannot continue with some tables inaccessible. Therefore, SAP HANA lets the initial master continue.

hdbnsutil

Some actions, supported by the hdbnsutil executable, access the persistence while the system is stopped. To avoid data corruption caused by unexpected active or reviving services, this program also checks for active name servers with network and storage based heartbeats and uses fencing to set the SCSI-3 persistent reservation.

SAN storages: After stopping hdbnsutil (or the name server), the SCSI-3 persistent reservations are intentionally not released. This ensures that no other service unintentionally accesses a persistence, such as still-running services on other hosts after a split-brain situation

Host Auto-failover Duration

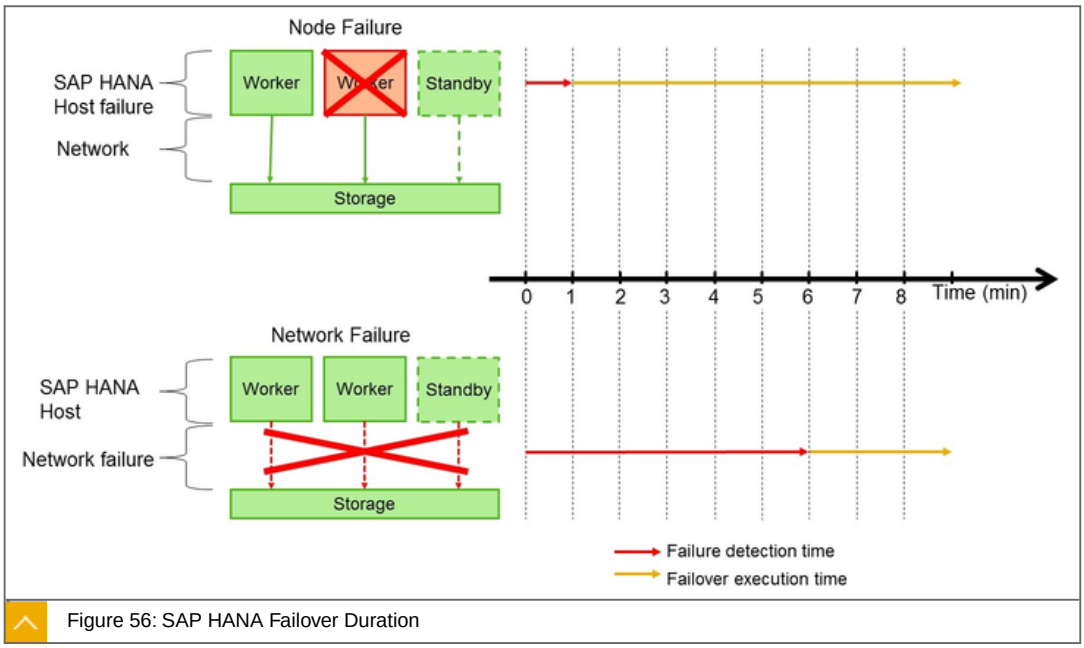


Figure 56: SAP HANA Failover Duration

The failover phase can be split into the following steps:

1. Failure detection

   Several watchdogs, retries, and timeouts are involved. Based on the failure condition, the detection time can vary, for example as follows:

   SAP HANA instance terminated or host shut down

   The checking host immediately gets errors from the OS layer and typically detects the failure in less than one minute.

   Network split

   The checking host must wait until the network times out, so failure detection typically takes three to six minutes. The timeouts could be reduced, but this is not recommended, as it would not allow recovery from short network outages, or could lead to a false failover decision in the case of heavy system load, where pings can take longer.

2. Failover execution

   The failover time is comparable to the time required for SAP HANA startup, because the services on a standby host are initially started, but run idle. During failover, they do the same initialization and persistence load as in regular service startups.

Host Start Order/Landscape Restart

All hosts can be started concurrently. The master name server candidates have different priorities as indicated by the role name MASTER 1, 2, and 3. The first master candidate becomes the active master. The index server roles, host roles, and storage partitions are reset, meaning that all configured worker hosts are used as worker again, even if the landscape was in a failed over state before shutdown.

Up to SPS11, if a host was previously used as a worker, its storage partition is kept as is, to avoid inefficient access patterns in clustered file systems. So over time, the storage partitions may have different sorting compared to their initial state after installation.

As of SPS12, a landscape restart considers the configured storage partitions, bringing the system back to its original state. As a prerequisite, the master name server must be on its original host (with a configured storage partition of 1). In an SAP HANA system replication setup, only the primary system performs this failback operation.

Failback

When a failover was performed and the failed host is available again, no automatic failback happens; the host starts as a standby. A controlled failback can be performed by stopping or restarting the configured standby host which, after a previous failover, is actually a worker. Automatic failback only happens when the complete landscape is restarted.

Master Nameserver Candidates

The initial host is a master candidate and the first two hosts added to a landscape become master candidates. When a standby host is added and none of the master candidates is a standby host, the last master candidate is moved to the new standby host. Having a standby host in the master candidate list allows faster master host failover because it avoids the previously-mentioned double failover.

Failover Groups

During installation and with SAP HANA Studio, a failover group can be configured per host. If a failover target host is available in the same group, it is preferred over hosts from other groups. This can be used to achieve better 'locality' in large systems, to use network/storage connections with less latency. When the parameter nameserver.ini/[failover]/ cross_failover_group is set to false, failover is restricted to hosts in the same group. This can be used to separate differently sized hardware or separate storage entities.

Application Configuration

In the connection information for SAP HANA SQL client libraries (for example, hdbuserstore), you can configure multiple host names. All master name server candidates should be configured there. The master candidates can be found using the following SQL statement:

```
select HOST from SYS.M_LANDSCAPE_HOST_CONFIGURATION where
NAMESERVER_CONFIG_ROLE like 'MASTER%' order by NAMESERVER_CONFIG_ROLE
```

Application Error Handling

Failover is not seamless. Errors during a failure phase are returned to the clients. Neither server nor client libraries have built-in 'retry' logic. Applications must be prepared and should try to reconnect.

- Master host failure: The client typically gets error -11312 (Connection to database server lost; check server and network status [System error: ...])
- Slave host failure: Basically, any error code can happen, because the master connection is still available, but some tables are no longer accessible and statements can fail at various steps.

LESSON SUMMARY
You should now be able to:

- Understand what happens during a failure of the master node

Unit 2
Lesson 8

Removing a Host from a Scale-Out System

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Remove a host from a scale-out system

Removal of a Host from a Scale-Out System

Business Example

As a SAP HANA database administrator you need to understand how to change the SAP HANA multi-host configuration. To better understand this feature, you remove the standby node from the SAP HANA system.

Remove a Host in a Multi-Host SAP HANA System

In a distributed SAP HANA system, tables and table partitions are assigned to an index server on a specific host at their time of creation, but this assignment can be changed. In certain situations, it is even necessary. You can use SAP HANA cockpit 2.0 together with the SAP HANA Database Explorer to execute automatic redistribution operations.

There are several occasions when tables or partitions of tables need to be moved to other servers. For example, if you plan to remove a host from your system, then you first need to move all the data on that host to the other hosts in the system. Redistributing tables may also be useful if you suspect that the current distribution is no longer optimal.

Although it is possible to move tables and table partitions manually from one host to another, this is neither practical nor feasible for a large-scale redistribution of data.

Redistribute Tables Before Removing a Host

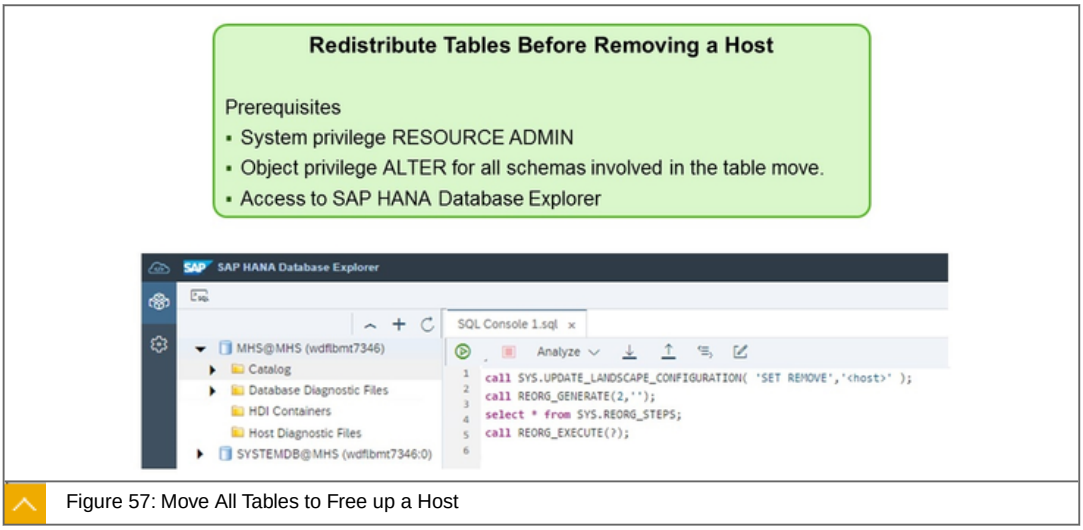


Figure 57: Move All Tables to Free up a Host

Before you can remove a host from your SAP HANA system, you must move the tables on the index server of the host in question to the index servers on the remaining hosts in the system.

Prerequisites

To redistribute tables across the hosts in your system, you must have the system privilege RESOURCE ADMIN and at least the object privilege ALTER for all schemas involved. As redistributing data is a critical operation, it is also recommended that you have saved the current distribution so you can restore it if necessary.

Procedure

1. From the SAP HANA Cockpit 2.0 – Database Directory screen, choose the Open SQL Console link.
2. In the Database Explorer, select your <Tenant>@< SID > and click on the Open SQL Console icon in the top-left corner.
3. In the SQL Console, execute the following commands:

```
call SYS.UPDATE_LANDSCAPE_CONFIGURATION( 'SET REMOVE','<host>' );
call REORG_GENERATE(2,'');
select * from SYS.REORG_STEPS;
call REORG_EXECUTE(?);
```

4. In the Host Failover screen, check the status of the host in the column Remove Status. If there is no column Remove Status, then add it using the Settings options.
5. If the column Remove Status has the value REORG FINISHED or REORG NOT REQUIRED, the host can be removed from the system.
6. Use the resident hdblcm, hdblcmgui, or the web-based hdblcm tool to remove the host from the multi-host SAP HANA system.

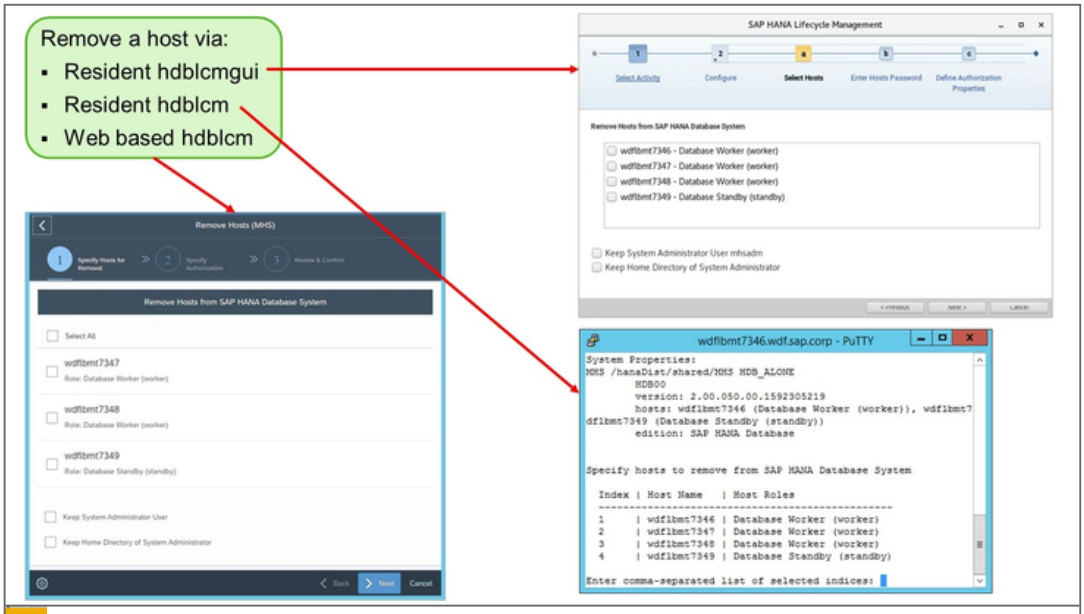


Figure 58: Tools to Remove a Host

Caution:
Removing a host 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.

Remove Hosts Using the Graphical User Interface or the Command-line Interface

You can remove hosts from an SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM) in the graphical user interface or the command-line interface.

General Prerequisites

- You have the credentials of the root user and the <sid>adm user.
- The SAP HANA system has been installed with the SAP HANA database lifecycle manager.
- The <sid>adm user has read and execute permissions for the directory that contains the installation medium.

If you want to remove a host that runs the master name server, another host that will take over the role of the master name server must be up and running.

Remove Hosts Using the Web User Interface

You can remove hosts from an SAP HANA system using the SAP HANA database lifecycle manager web user interface.

General Prerequisites:

- You require the credentials of the root user and the <sid>adm user.
- The SAP HANA system has been installed with the SAP HANA database lifecycle manager.
- The <sid>adm user has read and execute permissions for the directory that contains the installation medium.
- Communication port 1129 is open for SSL communication with the SAP Host Agent.

If you want to remove a host that runs the master name server, another host that will take over the role of the master name server must be up and running.

Web Browser Prerequisites:

On Microsoft Windows:

- Internet Explorer - Version 9 or higher

  If you are running Internet Explorer version 9, ensure 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

On SUSE Linux:

- Mozilla Firefox with XULRunner 10.0.4 ESR

On 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 the sap.m library in the SAPUI5 Developer Guide.

The Exercise Remove Host Explained

In the exercise for this lesson, the standby host is removed from the multi-host SAP HANA system. This action is performed by participant 03. In the next exercise we add the host again as a slave node.

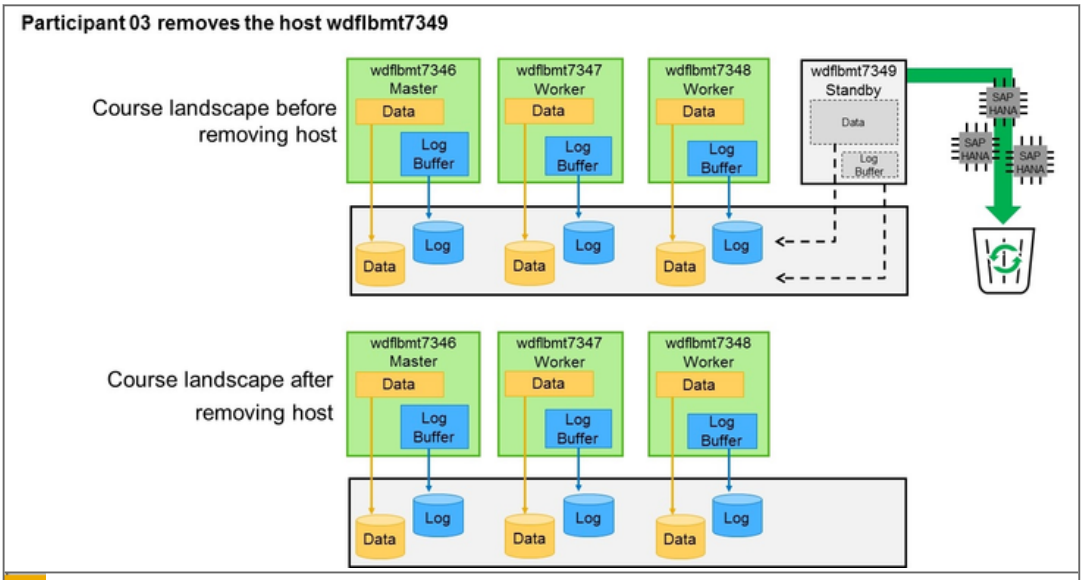


Figure 59: Exercise Remove Host Explained

During the exercises, in the Unit “Scale-out and Multitenant Database Containers”, this fourth server is used as an additional slave node.

LESSON SUMMARY
You should now be able to:

Remove a host from a scale-out system

Unit 2
Lesson 9

Adding a Host to a Scale-Out System

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Add a host to a scale-out system

Adding a Host to a Scale-Out System

Business Example

As an SAP HANA database administrator, you need to understand how to extend an SAP HANA multi-host system with additional hosts. To better understand this feature, you add a new slave node to the existing SAP HANA scale-out system.

Adding Hosts to an SAP HANA System

You can add hosts to an SAP HANA system using the SAP HANA database lifecycle manager (HDBLCM) resident program or the SAP HANA database lifecycle manager web user interface.

If you want to configure a new multiple-host (distributed) system during installation, see the multiple-host system installation information in the SAP HANA Server Installation and Update Guide.

Before adding a host to an SAP HANA system, you need to consider the following:

- If you are adding hosts from a host that is already integrated in the SAP HANA system
- If the system is a single-host or multiple-host system
- The number of hosts you want to add to the system at one time

If you are adding a host to a single-host system, the listen interface is automatically configured to global during the host addition. After the host is added to the system, the internal network address can be defined and the inter-service communication can be reconfigured to a different setting, if required.

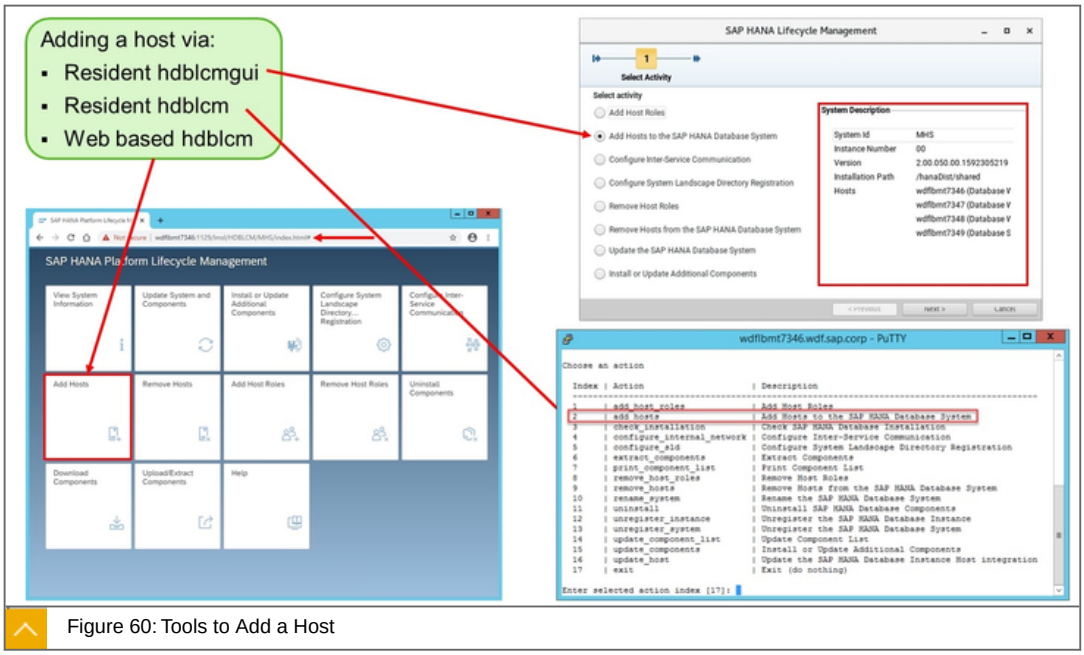


Figure 60: Tools to Add a Host

Add Hosts Using the Graphical User Interface or the Command-line Interface

You can add hosts to an SAP HANA system using the SAP HANA database lifecycle manager resident program in the graphical user interface.

Prerequisites:

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.

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. Ensure 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 resident program does not modify the properties of any existing user or group.

Add Hosts Using the Web User Interface

You can add hosts to an SAP HANA system using the SAP HANA database lifecycle manager web user interface.

Prerequisites:

On the host that is to be added, the SAP Host Agent is installed with SSL configured. The SAP Host Agent creates the <sapsys> group, if it does not exist prior to installation. Ensure that the group ID of the <sapsys> group is the same on all hosts.

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. Ensure 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) does not modify the properties of any existing user or group.

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.

Communication port 1129 is open.

Port 1129 is required for the SSL communication with the SAP Host Agent in a standalone browser using HTTPS.

Web Browser Prerequisites:

On Microsoft Windows:

Internet Explorer - Version 9 or higher

If you are running Internet Explorer version 9, ensure 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

On SUSE Linux:

Mozilla Firefox with XULRunner 10.0.4 ESR

On 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 the sap.m library in the SAPUI5 Developer Guide.

Redistribute Tables After Adding a Host

After you have added a new worker host to your SAP HANA system, you need to redistribute the tables in the system to balance the memory footprint of the tables and to improve performance (load balancing).

You can run table redistribution from the command line. This approach offers additional functionality including the option to modify, at runtime, some of the configuration parameters that control redistribution.

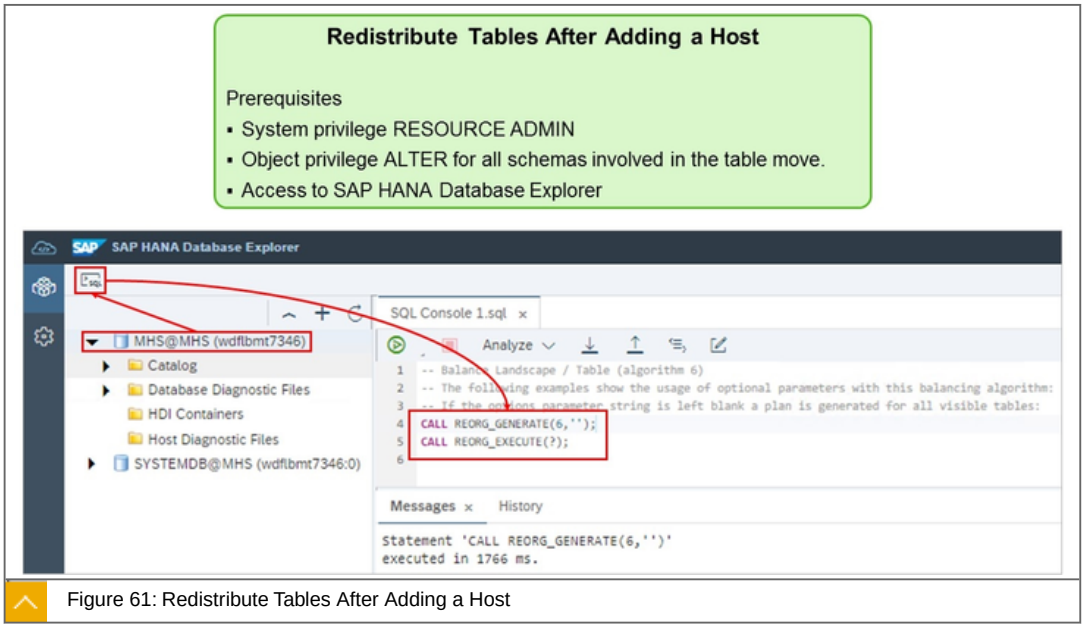


Figure 61: Redistribute Tables After Adding a Host

Table redistribution is based on the table placement rules defined in the table TABLE_PLACEMENT. These determine, for example, table sizes, partitioning threshold values, and preferred partition locations. Redistribution is a two-stage process: the first is to generate the plan, and the second is to execute the plan. Separate commands are used for each stage:

1. The plan generation command is a multi-purpose tool that requires an algorithm number as a parameter to determine which actions are executed. Depending on the algorithm selected, additional optional parameter values may also be available to give more control over the execution.
2. The plan execution command takes a single parameter which is the numeric plan ID value. You can retrieve this value (REORG_ID) from the REORG_OVERVIEW system view. Refer to the System Views section following.

The syntax for these commands is:

CALL REORG_GENERATE(< algorithm integer>, < optional parameter string>);

CALL REORG_EXECUTE(< plan_id>)

Resource admin privilege is required to call REORG_GENERATE(). The command only operates on tables and partitions that the executing user is allowed to see as catalog objects.

Generating the Plan: Algorithms and Options

The following table gives an overview of the most commonly-required algorithms and a summary of the options available for each one. See the examples and details of the options that follow.

Algorithm Number	Algorithm Name	Description
6	Balance landscape	This function checks if tables in the landscape are placed on invalid severs according to the table placement rules, and checks if a split or merge is necessary to achieve optimal positions for the partitions and tables and to evenly distribute tables across the index server hosts. Options: SCHEMA_NAME | TABLE_NAME | GROUP_NAME | GROUP_TYPE | GROUP_SUBTYPE | RECALC | NO_PLAN | NO_SPLIT | SCOPE
1	Add server	Run this check after adding one or more index servers to the landscape. If new partitions can be created, a plan is generated to split the tables and move the new partitions to the newly added index servers. Options: SCHEMA_NAME | TABLE_NAME | GROUP_NAME | GROUP_TYPE | GROUP_SUBTYPE | RECALC | NO_PLAN
4	Save	Save the current landscape setup. No optional parameters.
5	Restore	Restore a saved landscape setup. Enter the plan ID value as the optional parameter value.
7	Check number of partitions	This function checks if partitioned tables need to be re-partitioned and creates a plan to split tables if the partitions exceed a configured row count threshold. No optional parameters.
14	Check table placement	Check the current landscape against table placement rules and (if necessary) provide a plan to move tables and partitions to the correct hosts. Options: LEAVE_UNCHANGED_UNTOUCHED | KEEP_VALID | NO_SPLIT
15	Rerun plan	Rerun failed items from previously executed plans. Option: RERUN_ALL
16	Housekeeping	Perform housekeeping tasks. Additional privileges may be required for specific actions. Options: OPTIMIZE_COMPRESSION | DEFRAG | LOAD_TABLE | MERGE_DELTA | ALL

Optional Parameters

The following table gives more details of the optional parameters that are available.

Option	Type	Detail
SCHEMA_NAME	String	Restrict redistribution to the named schema(s) - comma-separated list.
TABLE_NAME	String	Restrict redistribution to the named table(s) - comma-separated list.
GROUP_NAME	String	Restrict redistribution to the named group(s) - comma-separated list.
GROUP_TYPE	String	Restrict redistribution to the named group types(s) - comma-separated list.
GROUP_SUBTYPE	String	Restrict redistribution to the named group sub types(s) - comma-separated list.
RECALC	True / False	If true, recalculate the landscape data of the last REORG_GENERATE run. This option works only if REORG_GENERATE has been called previously within the same connection session. This parameter can be used to speed up plan generation with different parameters.
NO_PLAN	True / False	If true, the planning stage of generating the plan is skipped. This can be used with external tools when landscape data needs to be collected and a distribution must be calculated, but may be modified.
SCOPE	Keyword	Scope the redistribution to include only the named items specified by the following keywords. The default value is 'ALL' so that all tables visible to the user are included in the redistribution. LOADED - Tables that are loaded or partially loaded UNLOADED - Tables that are not loaded FILLED - Tables with a record count greater than 10 EMPTY - Tables with a record count less than or equal to 10 USED - Tables with a total execution count greater than 10 UNUSED - Tables with a total execution count of less than or equal to 10 LOB - Tables with LOB columns NOLOB - Tables without LOB columns

Examples

Add server (algorithm 1):

With this algorithm, you can use the optional filter parameters to, for example, restrict redistribution to specified schemas, tables, table groups, and so on. The following example uses the SCHEMA_NAME option to generate a plan for all tables in schema SAPBWP:

```
CALL REORG_GENERATE(1, 'SCHEMA_NAME => SAPBWP')
```

Balance Landscape / Table (algorithm 6):

The following examples show the usage of optional parameters with this balancing algorithm.

If the options parameter string is left blank, a plan is generated for all visible tables:

```
CALL REORG_GENERATE(6,'');
```

This example uses the GROUP_NAME option to generate a plan for all tables in the three specified groups:

```
CALL REORG_GENERATE(6,'GROUP_NAME=>TABLEGROUP1, TABLEGROUP2, TABLEGROUP3');
```

This example uses the SCHEMA_NAME option to generate a plan for all tables in the schema SAPBWP:

```
CALL REORG_GENERATE(6,'SCHEMA_NAME => SAPBWP');
```

This example shows usage of the SCOPE option. The plan is restricted to only tables with a record count greater than 10 and that have no LOB columns:

```
CALL REORG_GENERATE(6, 'SCOPE=>FILLED,NOLOB');
```

System Views

The following system views show details of table redistribution. The last two views in the list show information about the most recent distribution operation. The details are deleted when the current connection to the database is closed.

REORG_OVERVIEW – Provides an overview of landscape redistributions.

REORG_STEPS – Shows details of the individual steps (items) of each plan.

REORG_PLAN – Contains details of the last table redistribution plan generated with this database connection.

REORG_PLAN_INFOS – Shows details (as key-value pairs) of the last executed redistribution (algorithm value and parameters used).

Exercise Add a Host Explained

In the exercise of this lesson, the empty host is added to the multi-host SAP HANA system as a slave node. This action is performed by participant 02.

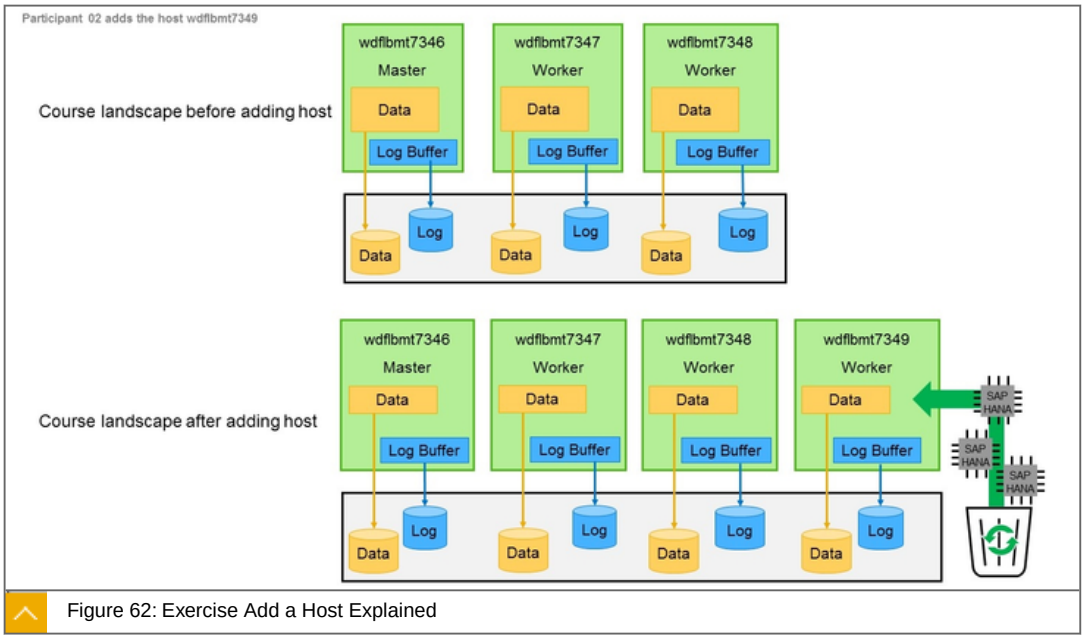


Figure 62: Exercise Add a Host Explained

During the exercises in the Unit “Scale-out and Multitenant Database Containers”, this additional slave node is used to store some of the created tenants.

LESSON SUMMARY
You should now be able to:

- Add a host to a scale-out system

Learning Assessment

1. Additional hosts can be added to a single-host SAP HANA database using the resident HDBLCM tools.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

2. Which of the following role types can be selected during the installation of an SAP HANA scale-out system?

   Choose the correct answers.

   - ☐ **A** Worker
   - ☐ **B** Master
   - ☐ **C** Slave
   - ☐ **D** Standby

3. Scale-up can be used to improve SAP HANA high availability.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

4. Which partitioning best practices should you use to create an optimal partitioning plan?

   Choose the correct answers.

   - ☐ **A** As few partition key columns as possible.
   - ☐ **B** SAP BW/4HANA: All partitions on same host.
   - ☐ **C** As many partitioned tables as possible.
   - ☐ **D** No additional unique constraints.

5. Which SAP HANA system views provide details of table redistribution?

   Choose the correct answers.

   - ☐ **A** M_REORG_ALGORITHMS
   - ☐ **B** EXPLAIN_PLAN_TABLE
   - ☐ **C** REORG_PLAN
   - ☐ **D** REORG_OVERVIEW

6. The Master 1 role is fixed after installation, and cannot be reconfigured to a different host.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

7. Which of the following are valid role types that you can assign to the name server and index server when you reconfigure the system in the Host Failover application?

   Choose the correct answers.

   - ☐ **A** Name server - Worker
   - ☐ **B** Name server - Master 1
   - ☐ **C** Index server - Master
   - ☐ **D** Index server - Standby

8. SAP HANA host auto-failover only handles slave node failures. Master node failures need to be handled by SAP HANA system replication.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

9. Which of the following are capabilities of SAP HANA host auto-failover?

   Choose the correct answers.

   - ☐ **A** STONITH is activated at the host level.
   - ☐ **B** Failover is performed at the SAP HANA services level.
   - ☐ **C** The failover happens automatically as an integral feature of SAP HANA.
   - ☐ **D** Data consistency is a key requirement.

10. When an SAP HANA master node fails, there is always a double failover.

    Determine whether this statement is true or false.

    - ☐ True
    - ☐ False

11. What configuration setting needs to be set up to avoid a double failover?

    Choose the correct answer.

    - ☐ **A** Set the third node to the Standby role.
    - ☐ **B** Have a standby host in the master candidate list.
    - ☐ **C** Have a worker host in the master candidate list.
    - ☐ **D** Assign the Worker role to the Master 3 nameserver node.

12. When removing a host from a multi-host SAP HANA system, you need to redistribute the tables first.

    Determine whether this statement is true or false.

    - ☐ True
    - ☐ False

13. Which of the following tools can you use to remove a host from a multi-host SAP HANA system?

    Choose the correct answers.

    - ☐ **A** Resident hdblcm
    - ☐ **B** hdblcmgui
    - ☐ **C** SAP HANA cockpit
    - ☐ **D** Web-based hdblcm

14. When a new host is added to an SAP HANA database system, the existing tables are automatically redistributed over the available nodes.

    Determine whether this statement is true or false.

    - ☐ True
    - ☐ False

Learning Assessment - Answers

1. Additional hosts can be added to a single-host SAP HANA database using the resident HDBLCM tools.

   Determine whether this statement is true or false.

   [X] True

   [ ] False

   You are correct! You can use the resident HDBLCM tools to add additional hosts to a SAP HANA database.

2. Which of the following role types can be selected during the installation of an SAP HANA scale-out system?

   Choose the correct answers.

   [X] **A** Worker

   [ ] **B** Master

   [ ] **C** Slave

   [X] **D** Standby

   You are correct! The roles Worker and Standby can be selected during the installation of an SAP HANA scale-out system.

3. Scale-up can be used to improve SAP HANA high availability.

   Determine whether this statement is true or false.

   [ ] True

   [X] False

   You are correct! Scale-up means increasing the size of one physical machine by increasing the amount of RAM available for processing. It does not improve SAP HANA high availability. Read more about this in the lesson "Explaining the SAP HANA High Availability Features" of the course HA201.

4. Which partitioning best practices should you use to create an optimal partitioning plan?

   Choose the correct answers.

   - [X] **A** As few partition key columns as possible.
   - [ ] **B** SAP BW/4HANA: All partitions on same host.
   - [ ] **C** As many partitioned tables as possible.
   - [X] **D** No additional unique constraints.

   You are correct! As few as possible partition key columns and no additional unique constraints are partitioning best practices. Read more about this in the lesson "Partitioning Tables" of the course HA201.

5. Which SAP HANA system views provide details of table redistribution?

   Choose the correct answers.

   - [ ] **A** M_REORG_ALGORITHMS
   - [ ] **B** EXPLAIN_PLAN_TABLE
   - [X] **C** REORG_PLAN
   - [X] **D** REORG_OVERVIEW

   You are correct! The SAP HANA system views REORG_PLAN and REORG_OVERVIEW provide details of table redistribution.

6. The Master 1 role is fixed after installation, and cannot be reconfigured to a different host.

   Determine whether this statement is true or false.

   - [ ] True
   - [X] False

   You are correct! Even the Master 1 role can be assigned to a different server.

7. Which of the following are valid role types that you can assign to the name server and index server when you reconfigure the system in the Host Failover application?

   Choose the correct answers.

   - [ ] **A** Name server - Worker
   - [X] **B** Name server - Master 1
   - [ ] **C** Index server - Master
   - [X] **D** Index server - Standby

   You are correct! Name server - Master 1 and Index server - Standby are valid role types that you can assign to the name server and index server when you reconfigure the system.

8. SAP HANA host auto-failover only handles slave node failures. Master node failures need to be handled by SAP HANA system replication.

   Determine whether this statement is true or false.

   - [ ] True
   - [X] False

   You are correct! SAP HANA host auto-failover handles failures of master and slave nodes.

9. Which of the following are capabilities of SAP HANA host auto-failover?

   Choose the correct answers.

   - [ ] **A** STONITH is activated at the host level.
   - [ ] **B** Failover is performed at the SAP HANA services level.
   - [X] **C** The failover happens automatically as an integral feature of SAP HANA.
   - [X] **D** Data consistency is a key requirement.

   You are correct! Automatic failover and data consistency are capabilities of SAP HANA host auto-failover.

10. When an SAP HANA master node fails, there is always a double failover.

    Determine whether this statement is true or false.

    - [ ] True
    - [X] False

    You are correct! There is no double failover if one of the master candidates is located on a standby server.

11. What configuration setting needs to be set up to avoid a double failover?

    Choose the correct answer.

    - [ ] **A** Set the third node to the Standby role.
    - [X] **B** Have a standby host in the master candidate list.
    - [ ] **C** Have a worker host in the master candidate list.
    - [ ] **D** Assign the Worker role to the Master 3 nameserver node.

    You are correct! Having a standby host in the master candidate list avoids a double failover.

12. When removing a host from a multi-host SAP HANA system, you need to redistribute the tables first.

    Determine whether this statement is true or false.

    - [X] True
    - [ ] False

    You are correct! Before a host can be removed from an SAP HANA system, all the tables assigned to that host need to be redistributed to the other nodes.

13. Which of the following tools can you use to remove a host from a multi-host SAP HANA system?

    Choose the correct answers.

    - [X] **A** Resident hdblcm
    - [ ] **B** hdblcmgui
    - [ ] **C** SAP HANA cockpit
    - [X] **D** Web-based hdblcm

    You are correct! You can use the Resident hdblcm and the Web-based hdblcm tools to remove a host from a multi-host SAP HANA system.

14. When a new host is added to an SAP HANA database system, the existing tables are automatically redistributed over the available nodes.

    Determine whether this statement is true or false.

    - [ ] True
    - [X] False

    You are correct! After adding an additional host, you must manually start the table redistribution. Load balancing is only done automatically for new tables.

UNIT 3

SAP HANA Disaster Tolerance

UNIT OBJECTIVES

- Explain SAP HANA storage replication
- Explain SAP HANA system replication

Set up SAP HANA system replication

Create tenant databases in a system replication scenario

Perform a takeover on the secondary system

Set up Active/Active SAP HANA system replication

Set up SAP HANA system replication with Secondary Time Travel

Explain Zero Downtime Maintenance

Explain Multitier and Multitarget System Replication

Unit 3
Lesson 1

Explaining SAP HANA Storage Replication

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Explain SAP HANA storage replication

Storage Replication

SAP HANA supports disaster tolerance solutions on the basis of replication using the hard disk I/O subsystem. The disaster tolerance solution is based on replication mechanisms of the hard disk I/O subsystem. All the data that is written to the persistence (data volume and log volume) by the primary SAP HANA system is replicated to a second location (secondary). The SAP HANA instance of the second location is not active (cold standby).

The mirroring is offered at the storage system level. It is offered together with the appliance as a special offering by our partners. The hardware partner defines how this concept is finally realized with their operation possibilities.

Generally, write requests are replicated to the hard disk I/O subsystem synchronously. When there are long distances between locations, the latency times for writing the redo log may increase. This means that the synchronous replication can be used up to certain distances only. The maximum distance depends on the performance requirements of the SAP HANA system, the respective solution of the hardware partner, and the network configuration in the customer environment.

Some hardware partners support an asynchronous transfer of the write requests, while the consistency of the data and transactions within the SAP HANA database is ensured. During a takeover in this case, the changes that were made last may be lost. In some application scenarios, this loss can be accepted, but in other cases, it cannot.

For example, in an SAP HANA data mart solution, data is replicated from a source system to the SAP HANA database using SLT. If the SAP HANA database loses the data that was inserted last due to a failover, SLT does not transfer the missing data again. As a result, the data has to be reloaded to the tables of the SAP HANA database. In this environment, only a synchronous replication between the primary SAP HANA system and the secondary SAP HANA system is useful.

Solutions with asynchronous replication can be used for longer distances between the locations because the latency times are not significantly affected when writing the redo log.

You can expect an impact on performance for data changing operations as soon as the synchronous mirroring is activated. The impact depends heavily on various external factors like distance, connection between data centers, and so on. The synchronous writing of the log with the concluding COMMITs is crucial.

In an emergency situation, the primary data center is no longer available and you must initiate a process for the takeover. So far, many customers have requested a manual process here, but an automated process can also be implemented. This take-over process then ends the mirroring officially, mounts the disks to the already installed SAP HANA software and

instances, and starts the secondary database side of the cluster. If the host names and instance names on both sides of the cluster are identical, no further steps with HDBRENAME are necessary.

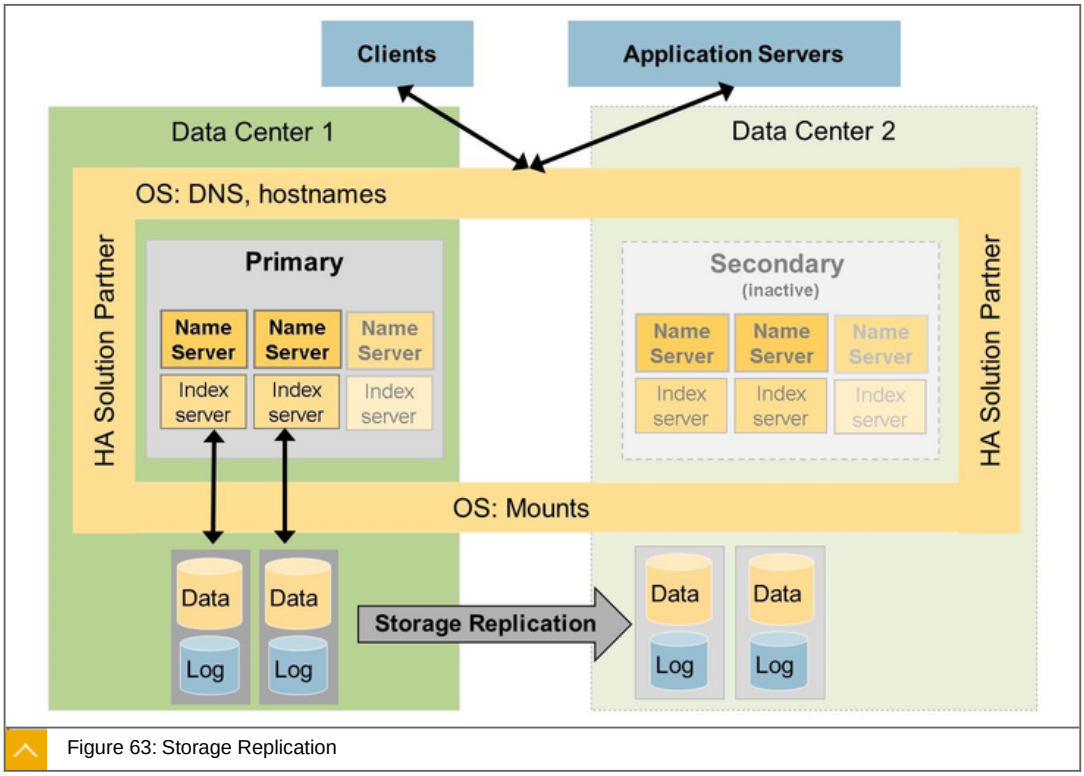


Figure 63: Storage Replication

Using Secondary Servers for Non-Production Systems

With SAP HANA storage replication, you can use the servers on the secondary system for non-production SAP HANA systems.

Depending on the hardware solution, you can use the servers of the secondary system for other SAP HANA systems (such as test systems) until the takeover. During a takeover, other running systems are switched off, the replication between the locations is interrupted, the mount points of the replicate are made available for the secondary servers, and the SAP HANA system is started. When using servers of the secondary system for other SAP HANA systems, these do not use the hard disk storage system that contains the replicate of the primary system.

You can run a development or QA instance of the three-tier installation on this secondary cluster hardware, simply to use it until the takeover is executed. The take-over then stops these development or QA instances and mounts the production disks to the hosts. It requires an additional set of disks for the development and QA instance.

The same applies to asynchronous mirroring solutions for distant data centers (> 100 km). Some hardware partners have concepts available to offer this asynchronous storage replication. For more information, see SAP Note: 1755396.

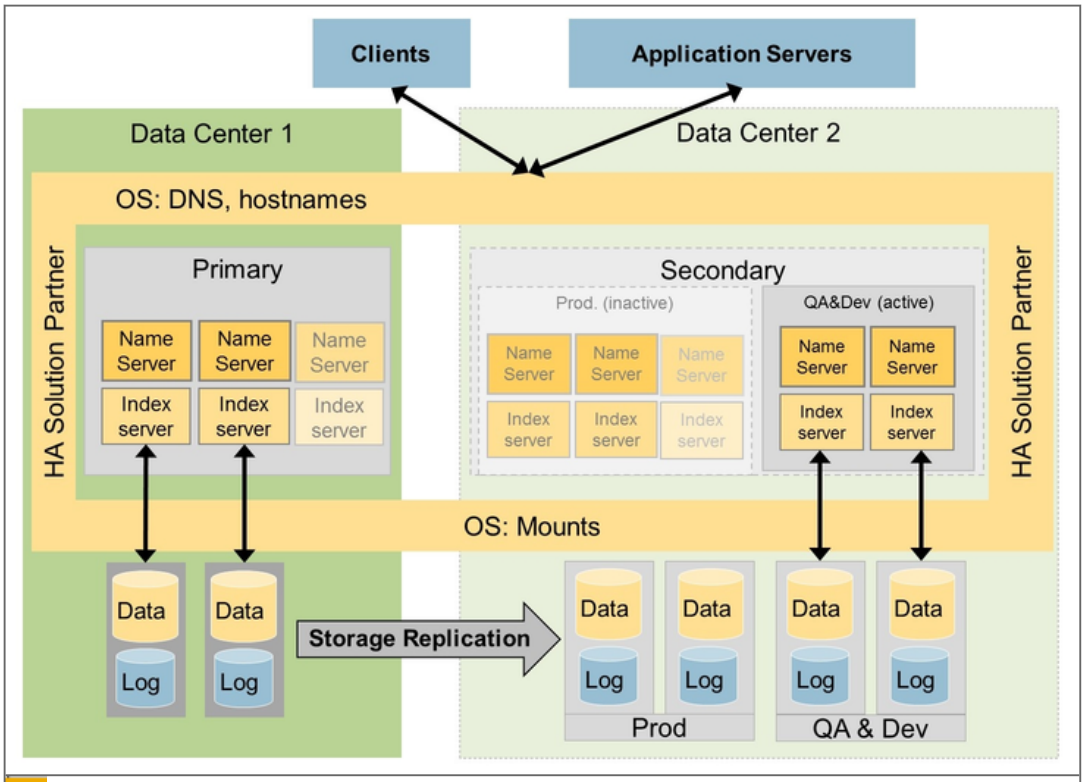


Figure 64: Storage Replication with QA and Development System on a Second Site

Supported Storage Solutions

Direct support by hardware vendor or partners are as follows:

Existing storage replication solution certifications for all SAP HANA appliances continue their validity (solutions reported in SAP Note: 1755396).

All newer solutions are supported directly by corresponding vendors (hardware or storage partners).

No further certification of these storage replication solutions from SAP is required for use with SAP HANA.

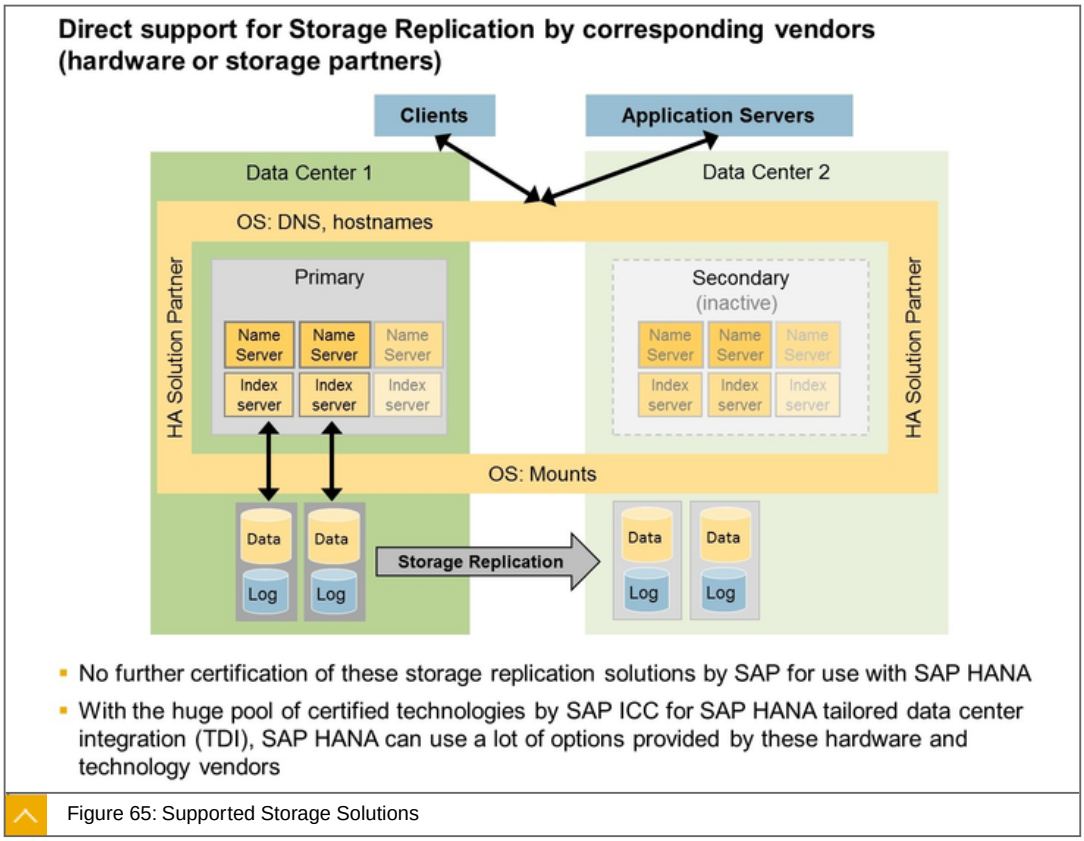


Figure 65: Supported Storage Solutions

LESSON SUMMARY
You should now be able to:

Explain SAP HANA storage replication

Unit 3
Lesson 2

Explaining SAP HANA System Replication

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Explain SAP HANA system replication

System Replication

Usually system replication is set up so that a secondary standby system is configured as an exact copy of the active primary system, with the same number of active hosts in each system. The number of standby hosts do not need to be identical.

With multitier system replication, you have one primary system and can have multiple secondary systems. Each service instance of the primary SAP HANA system communicates with a counterpart in the secondary system.

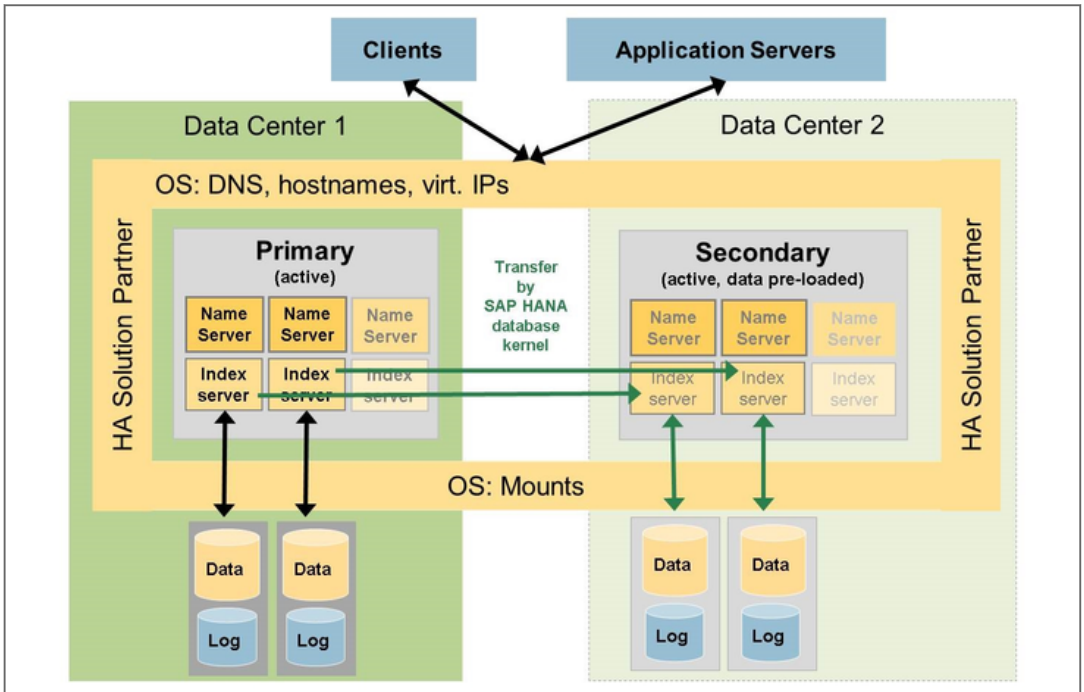


Figure 66: System Replication

The secondary system can be located near the primary system to serve as a rapid failover solution for planned downtime, or to handle storage corruption or other local faults. Alternatively, it can be installed in a remote site to be used in a disaster recovery scenario. Both approaches can be linked together with multitier system replication. Like storage replication, this disaster recovery option requires a reliable connection channel between the primary and secondary sites. The instances in the secondary system operate in recovery

mode. In this mode, all secondary system services constantly communicate with their primary counterparts.

A cluster across data centers with database controlled transfer is realized by system replication.

System replication has the following advantages:

Memory is continuously loaded on a secondary site in preparation for the possible takeover and occupies resources.

Switch-over is faster than with storage replication or mirroring (2-5 minutes).

There is a very short performance ramp (only minutes, not hours, without preparation).

System replication has the following disadvantages:

The hardware (memory and CPU) is actively used on the secondary site for the standby or shadow processes.

Replication Modes

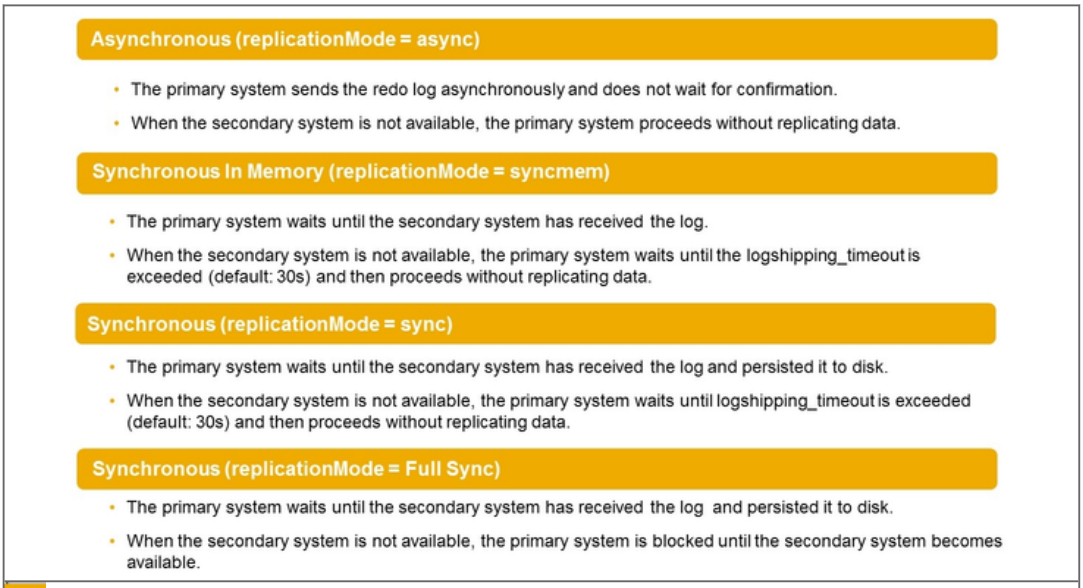


Figure 67: Replication Modes

When the secondary system is started in recovery mode, each service component establishes a connection with its counterpart, and requests a snapshot of the data in the primary system. From then on, all logged changes in the primary system are replicated. Whenever logs are persisted in the primary system, they are also sent to the secondary system. A transaction in the primary system is not committed until the logs are replicated. What this means can be configured by choosing one of the log replication modes:

Asynchronous: The primary system sends redo log buffers to the secondary system asynchronously. The primary system commits a transaction when it has been written to the log file of the primary system and sent to the secondary system through the network. It does not wait for confirmation from the secondary system.

This option provides better performance because it is not necessary to wait for log I/O on the secondary system. Database consistency across all services on the secondary system

is guaranteed. However, it is more vulnerable to data loss. Data changes may be lost on takeover.

Synchronous in-memory (default): The primary system commits the transaction after it receives a reply that the log was received by the secondary system, but before it has been persisted. The transaction delay in the primary system is shorter, because it only includes the data transmission time.

Synchronous: The primary system does not commit a transaction until it receives confirmation that the log has been persisted in the secondary system. This mode guarantees immediate consistency between both systems. However, the transaction is delayed by the time it takes to transmit the data to and persist it in the secondary system.

Synchronous with full sync: The log write is successful when the log buffer has been written to the log file of the primary and the secondary instance. In addition, when the secondary system is disconnected (for example, because of network failure), the primary systems suspend transaction processing until the connection to the secondary system is re-established. No data loss occurs in this scenario.

If the connection to the secondary system is lost, or the secondary system crashes, the primary system resumes replication after a brief, configurable, timeout. The secondary system persists, but does not immediately replay the received log. To avoid a growing list of logs, incremental data snapshots are transmitted asynchronously from time to time from the primary system to the secondary system. If the secondary system has to take over, only the part of the log needs to be replayed that represents changes that were made after the most recent data snapshot. In addition to snapshots, the primary system also transfers status information regarding which table columns are currently loaded into memory. The secondary system correspondingly preloads these columns. In the event of a failure that justifies full system takeover, an administrator instructs the secondary system to switch from recovery mode to full operation. The secondary system, which already preloaded the same column data as the primary system, becomes the primary system by replaying the last transaction logs, and then starts to accept queries.

Note the following for synchronous and asynchronous setups:

In a synchronous state, no committed transaction is lost. The open transaction is restarted and clients reconnect to SAP HANA for this. Synchronous setups are required for distances in the range 50-100 km.

In case of an asynchronous setup, there is some loss. This depends on the time period where the secondary site was not reachable or the line was too weak to cope with the data transfer quickly enough. These setups are used for longer distances, where the distance between data centers is 100 km or more. However, they also occur if the impact of the standby process is not allowed to feedback into daily operation (change performance).

Minimal Setup for System Replication

The minimal setup for system replication in one data center for fast takeovers is shown in the figure, Minimal Setup for System Replication.

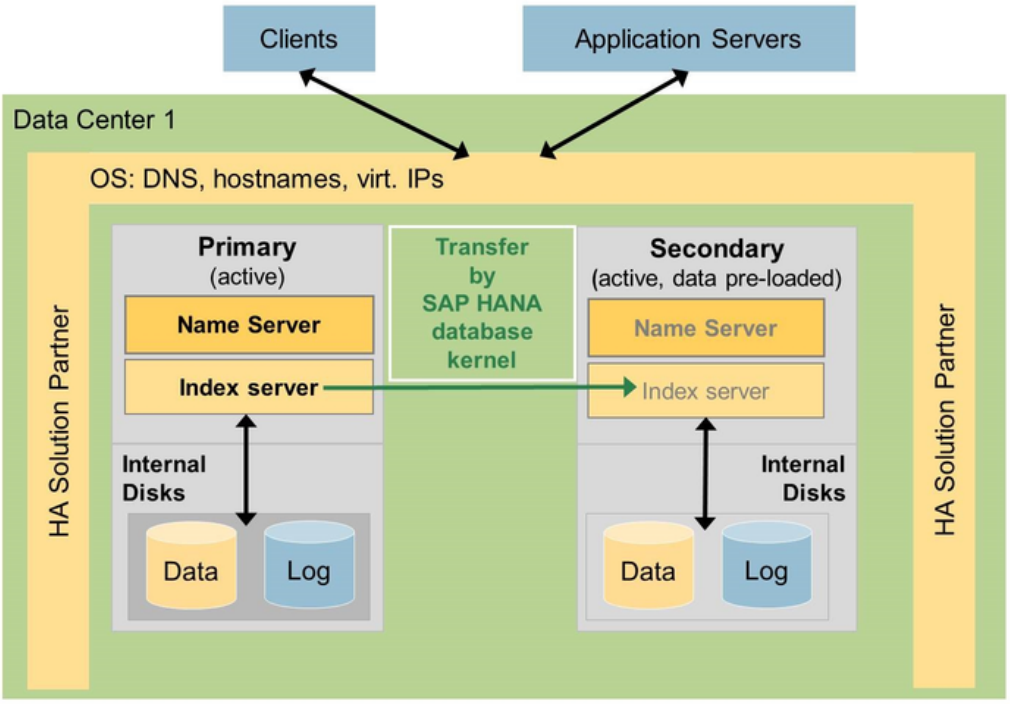


Figure 68: Minimal Setup for System Replication

Operation Modes for System Replication

There are three different operation modes for the configuration of system replication.

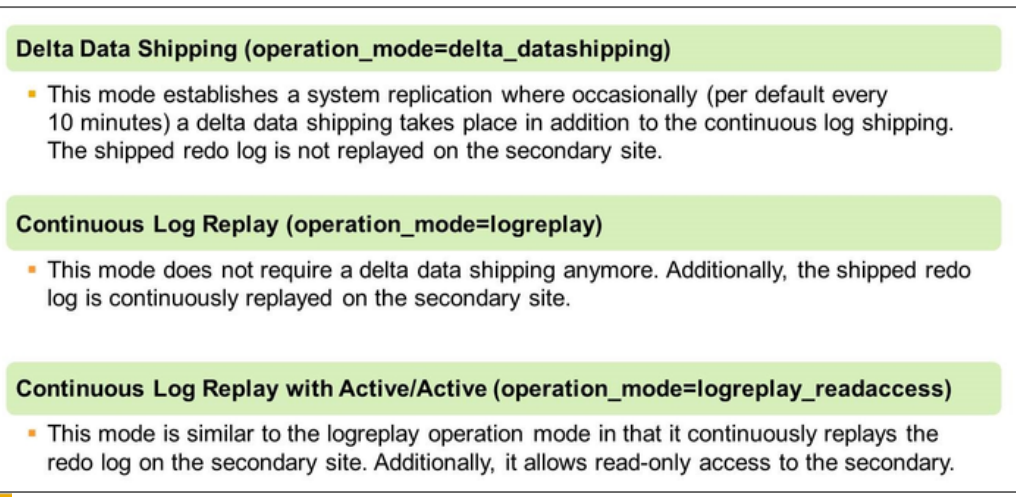


Figure 69: Operation Modes

Note:
A comparison between delta_datashipping and logreplay with regard to network traffic shows significantly reduced network traffic. Delta data shipping displays a peak every 10 minutes when delta data shipping is triggered, whereas logreplay shows continuously shipped log buffers.

System Replication with Operation Mode Delta Data Shipping

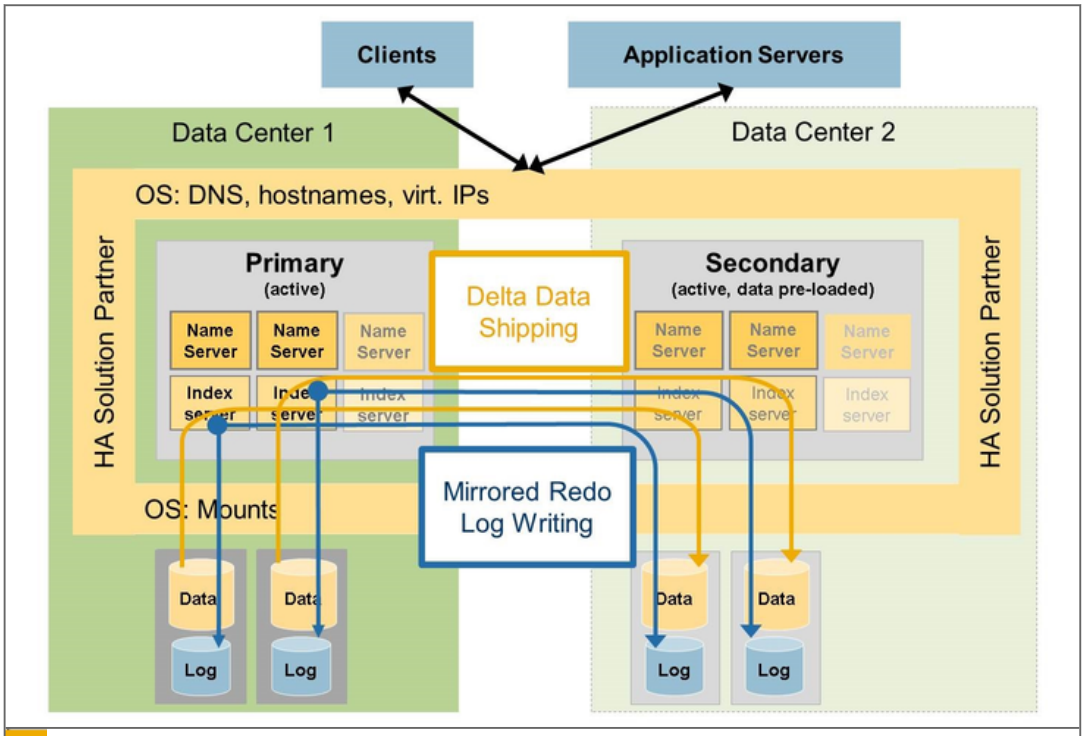


Figure 70: System Replication with Operation Mode Delta Data Shipping

The events for the start of a transport are as follows:

1. The primary system creates an internal data package similar to a full data backup and transfers this initially to the secondary site. The transport happens asynchronously.
2. Log information is transferred in parallel to the initial data transfer. The log is transported asynchronously until the commit of the finished transaction occurs. With the commit, all other log information that is not yet transferred or written, as well as the final commit, must also be written synchronously. This must occur before the primary productively-used database can continue transactional work.
3. All load and unload operations of the main indexes and table columns are monitored and offered with the incremental data transfer to the secondary system. These main indexes and table columns are then loaded or unloaded equivalently to memory in preparation for the takeover.

The events during incremental transport are as follows:

1. With the help of the shadow memory concept operation of SAP HANA, small incremental backups are transferred to the delta data package every 10 minutes at the secondary site. The default parameter setting is 600 seconds.
2. With this delta data information, information from the loaded main indexes into SAP HANA on the primary site are also transferred to the secondary site. This is to prepare the main memory with these main indexes on the secondary site too.

System Replication with Operation Mode Log Replay

Since the first version of system replication, the delta_datashipping operation mode has been the default replication method. With the logreplay operation mode, delta data shippings are no longer necessary. The takeover time has been reduced and more components are already initialized at replication time.

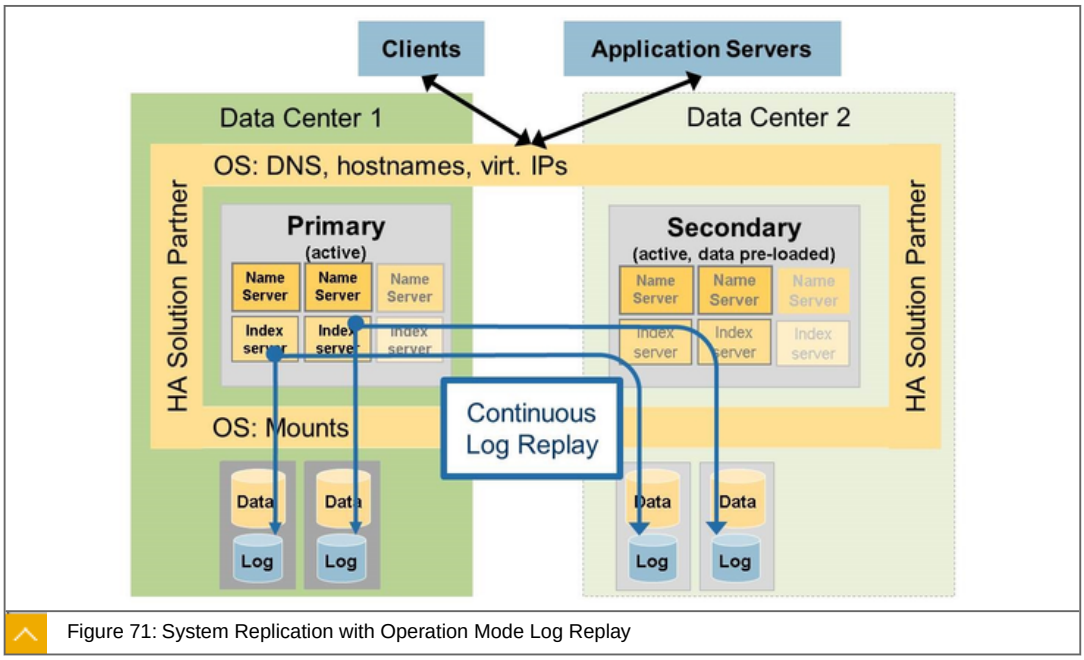


Figure 71: System Replication with Operation Mode Log Replay

In the logreplay operation mode, the system replication uses an initial data shipping to initialize the secondary site. After that, only log shipping is complete and log buffers received by the secondary are replayed there. Savepoints are executed individually for each service and column table merges are executed on the secondary site.

In the logreplay mode of operation, log segments can be marked as retained so that they can sync a secondary system after a disconnect.

With continuous log replay, delta data shipping cannot be used to sync a secondary site. This is because although the primary and secondary persistence are logically compatible, they are no longer physically compatible. This means that the data contained in the persistence is the same, but the layout of the data on pages can be different on the secondary site. Therefore, a secondary site can sync only using delta log shipping. This is relevant for the following situations:

- The secondary site has been disconnected for some time (for example, because of a network problem or temporary shutdown of the secondary site).
- A former primary site has been registered for failback.

The secondary site only uses the log in the online log area of the primary SAP HANA system for syncing. To sync the secondary site, the log must be retained for a longer time period than previously. If syncing using delta log shipping does not work, for example because the log has been reused, a full data shipping is necessary. To avoid this, the concept of log retention has been introduced.

System Replication with QA and Development System on the Secondary Site

It is possible to make use of the secondary site for running QA and development systems while the primary system is in production.

Prerequisites for System Replication with QA and Development System on the Secondary Site

The following prerequisites must be taken into account:

- Additional independent disk volume is needed for Development/QA systems. Because the secondary site requires the same I/O capacity as the primary site, the additional systems must not have a negative impact on the secondary's I/O. Therefore, it is recommended to have a separate storage infrastructure for each system.
- The SIDs and instance numbers have to be different for Development/QA. The <instance number>+1 of the productive system must not be used, but must be free on both sites, because this port range is used for system replication communication.
- Preload of tables must be switched off on the secondary site, using:

  global.ini/[system_replication]-> preload_column_tables=false
- The takeover process takes longer because no data is preloaded in memory at the secondary site (could still meet SLAs for disaster recovery).
- Development/QA systems need to be shut down in the case of a takeover.
- The global allocation limit on the secondary must be set in a way that the available memory covers the memory needed by the secondary system as well as the Development/QA systems, using:

  global.ini/[memorymanager]-> global_allocation_limit

The configured operation mode influences the memory size required on the secondary site as follows:

Operation Mode	Memory Needed on Secondary Site
delta_datashipping	row store size + 20 GB (minimum 64 GB)
logreplay	row store size + size of column tables loaded in memory + 50 GB

If the row store size grows during operation of the primary, it might become necessary to increase the global_allocation_limit on the secondary site. It is possible to change the global.ini on the secondary site accordingly and then activate the change with "hdbnsutil –reconfig" (because SQL is not possible in this state).

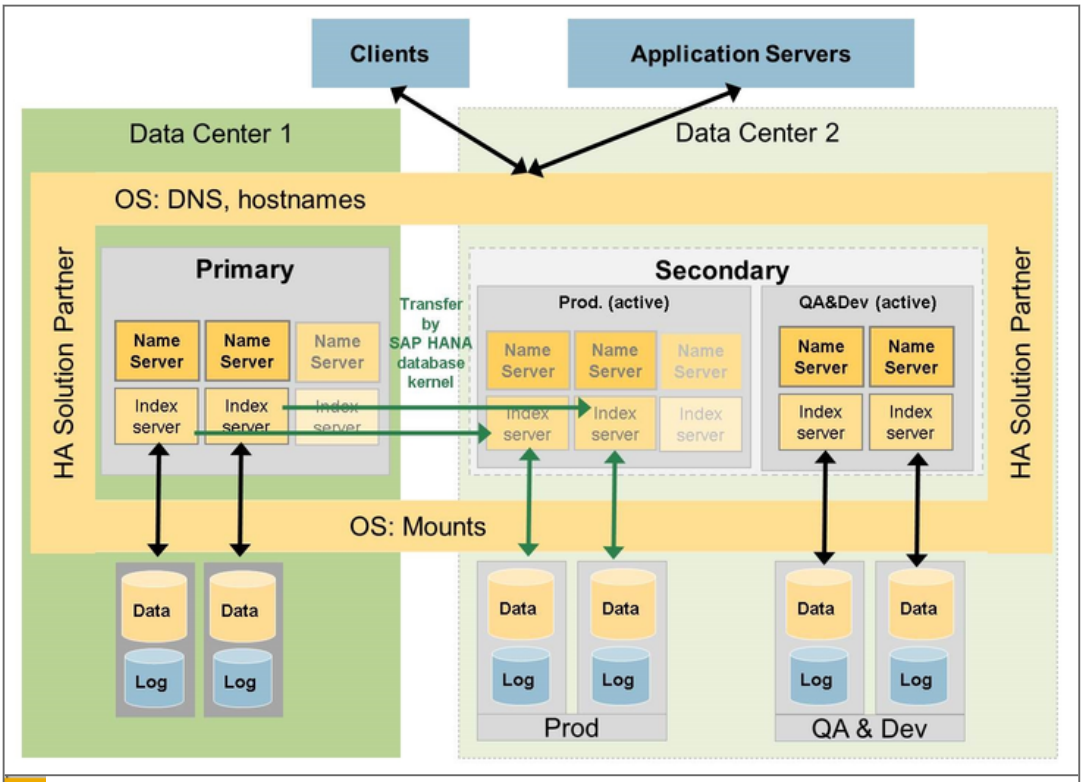


Figure 72: System Replication with QA and Development Systems on a Secondary Site

The advantages of system replication with a Development/QA system on a secondary site include the following:

- Development/QA operated on the secondary site (mixed cost calculation).
- Synchronous and asynchronous solution available.
- Impact of synchronous solution on the primary site is at about 10% (in contrast to about 25% with storage replication).
- The transfer process from primary to secondary is optimized and a lesser transfer amount is necessary compared to storage replication.
- During the takeover to the secondary site, only a roll forward is necessary because the latest data synchronization point is necessary.

The disadvantages of system replication with a Development/QA system on a secondary site include the following:

- Table and column data cannot be loaded continuously into memory on the secondary site.
- Hardware (memory and CPU) is actively used for Development/QA and partly for the standby or shadow processes.
- Takeover is similar to storage mirroring (20 to 30 minutes at best).
- Performance ramp is similar to storage mirroring (1 to 3 hours).
- QA and Development need their own disk infrastructure carefully separated so as not to have influencing effects on each other.

Additional Information	SAP Note
FAQ: SAP HANA System Replication	1999880
FAQ: SAP HANA Database Backup & Recovery in an SAP HANA System Replication Landscape	2165547
Collection of How-To Guides and Whitepapers For SAP HANA High Availability: ▪ FAQ High Availability for SAP HANA ▪ How To Perform System Replication for SAP HANA ▪ How To Configure Network Settings for HANA System Replication ▪ Network Required for SAP HANA system replication	2407186

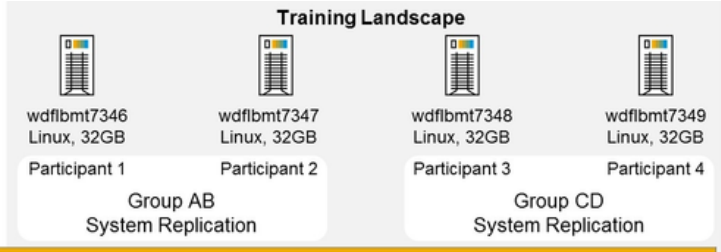
Figure 73: System Replication – Additional Information

Setup for the Exercises

In the following exercises, you set up the SAP HANA system replication. As a prerequisite, all participants have to install an SAP HANA system. In the next exercises, two participants work together as a team to enable system replication between their systems.

Participant 1 and 2 form a team named system replication group AB. Participant 3 and 4 form a team named system replication group BC.

There are two exercises to enable system replication. In the exercise “Set Up SAP HANA System Replication”, the main part is done by participant 1 and 3. In the other exercise “Set up Active/Active SAP HANA System Replication”, the main part is done by participant 2 and 4.



	System Replication Group AB		System Replication Group CD	
<Linux host>	wdflbmt7346	wdflbmt7347	wdflbmt7348	wdflbmt7349
All Participants: Install systems for system replication	<SID> = H10	<SID> = H10	<SID> = H20	<SID> = H20
Participant 01 and 03: Enable system replication	Primary	Secondary	Primary	Secondary
Participant 02 and 04: Enable system replication (active / active)	Primary	Secondary	Primary	Secondary

Figure 74: Setup System Replication

LESSON SUMMARY
You should now be able to:

Explain SAP HANA system replication

Unit 3
Lesson 3

Setting up SAP HANA System Replication

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Set up SAP HANA system replication

Overview of Configuration Steps

Configuration Steps to Set up SAP HANA System Replication

1. Start the primary system.
2. Create an initial data backup or a storage snapshot on the primary system.
3. Enable system replication on the primary system.
4. Prepare the secondary system for authentication by copying the system PKI SSFS .key and the .dat file from the primary system to the secondary system.
5. Register the secondary system and establish a connection between the secondary and primary systems.

The configuration tasks on the primary and secondary systems to set up system replication are shown in the figure Setup of System Replication. With this configuration, you can recover from a data center outage by switching to a secondary site. The primary system stays online during this procedure.

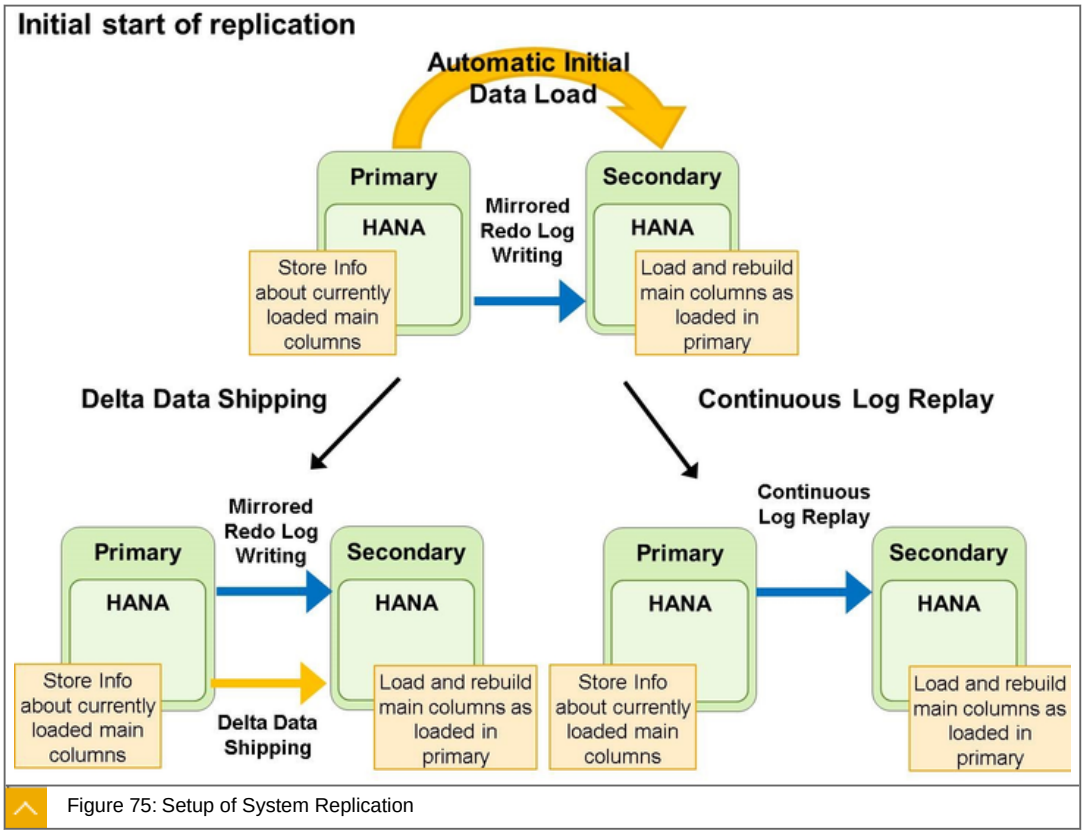


Figure 75: Setup of System Replication

The following steps are performed during the setup of system replication:

1. The primary system is informed to enable system replication.
2. The secondary database is stopped. Content is wiped out during the initial load, with a full data backup later during the initial start of replication.
3. The secondary system is advised to connect to the primary system, and communicates about the attempt to start the system replication standby process.

   This process is secured with certificates and so on.

   Only one command is needed: HDBNSUTIL

   Both sides must have the same number of active and standby hosts with the same sizing (memory and CPU).

   SAP HANA itself handles the relationships of, for example, scale-out setups on both sides (primary to secondary) and how communication is established with each counterpart.

   Communication takes place internally between sites on TREXnet.

Note:
If the primary connection between data centers is too weak for an initial data load (usually TBs), then use snapshot data backups for setting up SAP HANA system replication initialization.

Additional Configuration Steps to Enable SAP HANA System Replication

Starting with SAP HANA 2.0, additional configuration steps are required to set up SAP HANA system replication, because replication connections now use certificate-based authentication.

System replication with SAP HANA 2.0 requires authentication for the data and log shipping channels. The authentication is done using the certificates in the system PKI SSFS store. An additional manual setup step is required to exchange certificates in the system PKI SSFS store between primary and secondary sites. For more information, see SAP Note: 2369981.

Additional Configuration Steps to Enable SAP HANA System Replication

Copy the system PKI SSFS KEY and DAT files from the primary site to the secondary site.

The files can be found at the following locations:

- /usr/sap/<SID>/SYS/global/security/rsecssfs/data/SSFS_<SID>.DAT
- /usr/sap/<SID>/SYS/global/security/rsecssfs/key/SSFS_<SID>.KEY

For more information, see SAP Note 2369981 - Required configuration steps for authentication with HANA System Replication.

Note:
If you installed XS advanced, you must also copy the XSA SSFS .key and the .dat file from the primary system to the secondary system in the following directories:

/usr/sap/<SID>/SYS/global/xsa/security/ssfs/data/
SSFS_<SID>.DAT

/usr/sap/<SID>/SYS/global/xsa/security/ssfs/key/
SSFS_<SID>.KEY

For more information, see SAP Note 2300936 - Host Auto-Failover & System Replication Setup with SAP HANA extended application services, advanced model.

The copied files become active during system restart. Therefore, it is recommended to copy the files when the secondary SAP HANA system is offline, for example, before registration.

Enablement of SAP HANA System Replication

System replication can be set up or managed on the command line with hdbnsutil, using the SAP HANA cockpit, SAP HANA studio, or with SAP Landscape Management.

The following administration activities are possible with hdbnsutil, using the SAP HANA cockpit, or SAP HANA studio:

Performing the initial setup, that is, enabling system replication and establishing the connection between two identical systems.

Monitoring the status of system replication to ensure that both systems are in sync.

Triggering takeover by the secondary system in the event of a disaster and failback once the original system is available again.

Disabling system replication.

Enable SAP HANA System Replication Using SAP HANA Cockpit

There are two ways to set up SAP HANA system replication in the SAP HANA cockpit:

- Enable the primary system and then register the secondary system from the primary system in one configuration step.
- Enable system replication on the primary system and then register the secondary system in a second step.

The steps to configure the primary and the secondary system using SAP HANA cockpit are outlined in the figure, Enable System Replication.

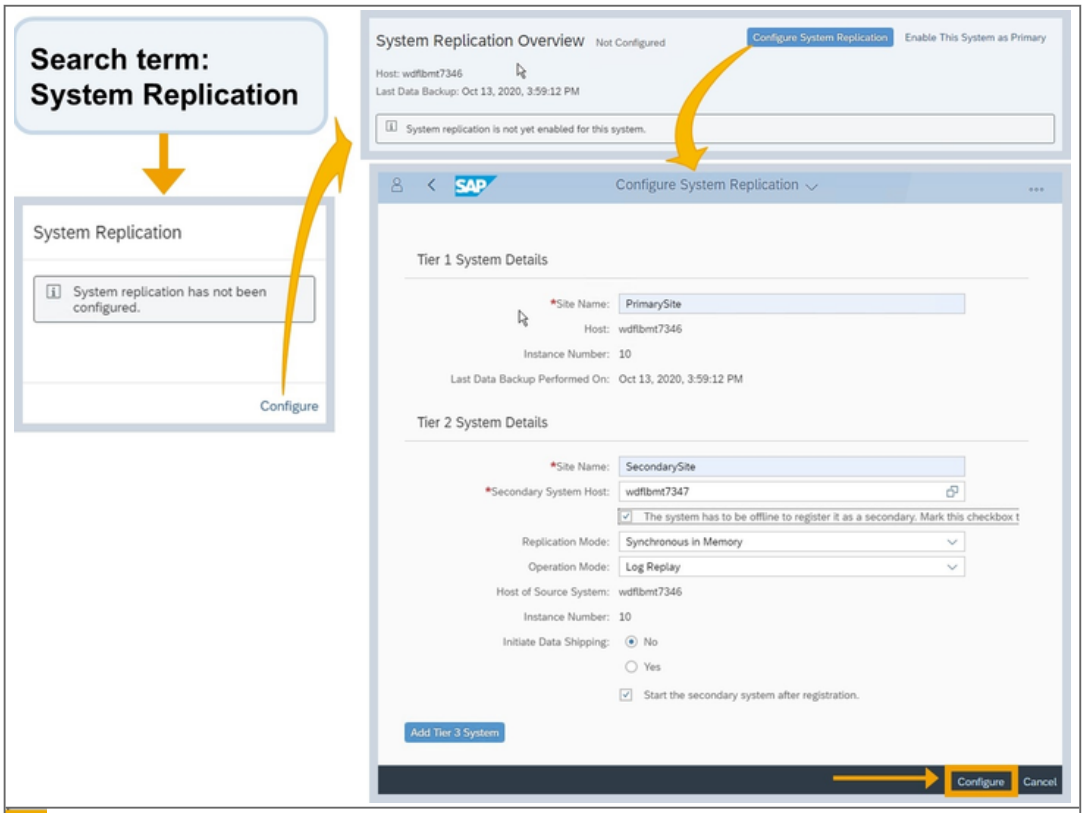


Figure 76: Enable System Replication

You have enabled system replication and registered the secondary system with the primary system. The secondary system operates in recovery mode. All secondary system services constantly communicate with their primary counterparts, replicate and persist data and logs, and load data to memory. However, the secondary system does not accept SQL connections.

To set up SAP HANA system replication between two identical SAP HANA systems, you must first enable system replication on the primary system and then register the secondary system.

Enable SAP HANA System Replication with hdbnsutil

It is also possible to configure SAP HANA system replication with the command line tool hdbnsutil as <sid>adm at the OS level. The command line tool can be a part of a script, which executes further steps beyond system replication.

Enable SAP HANA System Replication with hdbnsutil

1. Create a data backup of the primary system.
2. Enable the primary system and give the primary system a logical name:

   hdbnsutil -sr_enable --name=PRIMARY
3. Stop the secondary system:

   sapcontrol –nr <instance_number> -function StopSystem HDB
4. Register the secondary system (choose replication mode and operation mode):

```
hdbnsutil -sr_register --remoteHost=<primary hostname>
--remoteInstance=<instance number>
--replicationMode=<sync|syncmem|async>
--operationMode=<delta_datashipping|logreplay>
--name=SECONDARY
```

5. Start the secondary system to start replication:

   sapcontrol –nr <instance_number> -function StartSystem HDB

Once the secondary system is started, the replication process starts automatically.

Enable the Full Sync Option for SAP HANA System Replication

When activated, the full sync option for SAP HANA system replication ensures that a log buffer is shipped to the secondary system before a commit takes place on the local primary system.

Full Sync Option for SAP HANA System Replication

The full sync option can be enabled for SYNC replication (that is, not for SYNCMEM). With the full sync option activated, transaction processing occurs on the primary blocks. If the secondary system is not currently connected, the newly created log buffers cannot be shipped to the secondary site. This behavior ensures that no transaction can be locally committed without shipping the log buffers to the secondary site. The full sync option can be switched on and off using the command: hdbnsutil -sr_fullsync --enable|--disable

This command changes the setting of the enable_full_sync parameter in the system_replication section of the global.ini file accordingly. However, in a running system, full sync does not become active immediately. This is done to prevent the system from blocking transactions immediately when setting the parameter to true. Instead, full sync has to first be enabled by the administrator. In a second step, it is internally activated when the secondary is connected and becomes ACTIVE.

In the M_SERVICE_REPLICATION system view, the setting of the full sync option can be viewed in using SQL.

The full sync option can have the following values:

DISABLED: Full sync is not configured at all. The parameter enable_full_sync = false in the system_replication section of the global.ini file.

ENABLED: Full sync is configured, but it is not yet active, so transactions do not block in this state. To become active, the secondary has to connect and REPLICATION_STATUS must be ACTIVE.

ACTIVE: Full sync mode is configured and active. If the network connection to a connected secondary is closed, transactions on the primary side block in this state.

If full sync is enabled when an active secondary is currently connected, FULL_SYNC is immediately set to ACTIVE.

Caution:
If the secondary is stopped, disable FULL_SYNC. Otherwise, the primary blocks and it is not possible to stop it.

Note:
Resolving a blocking situation of the primary caused by the enabled full sync option must be done with the hdbnsutil command, because a configuration changing command could also block in this state. This is also necessary if you want to shut down the currently blocking primary. Otherwise, it is not possible to stop it.

Compression Methods for Log and Data Shipping

SAP HANA system replication supports a number of compression methods for log and data shipping.

The following types of compression for log and data shipping are supported:

Log

- Log buffer tail compression (by default)
- Log buffer content compression

Data

- Data page compression

Log buffer tail compression is turned on by default. All log buffers are aligned to 4 KB boundaries by a filler entry. With log buffer tail compression, the filler entry is cut off from the buffer before sending it over the network and added again when the buffer has reached the secondary site. So only the net buffer size is transferred to the secondary site.

The size of the filler entry is less than 4 KB. This is the maximum size reduction per sent log buffer. If the log buffers size is quite large, the compression ratio is quite limited.

Log buffer and page content compression can be activated by parameter settings.

Log buffers and data pages shipped to the secondary site can be compressed using a lossless compression algorithm (LZ4). By default, content compression is turned off. You can turn it on by setting the following configuration parameters on the secondary site in the system_replication section of the global.ini file.

Configuration Parameters to Activate Compression

Enable compression of a log when it is sent to the secondary site:

enable_log_compression = true

Enable compression of data when it is sent to the secondary site:

enable_data_compression = true

Note:
After changing these parameters, the secondary site needs to be reconnected to the primary site.

Log and data compression is especially useful when system replication is used over long distances, for example, using the ASYNC replication mode.

The open source compression algorithm LZ4 has been selected because of its speed and compression ratios, and the relatively low time overhead introduced for compression/ decompression. Log buffer content compression also works in combination with log buffer tail compression. Therefore, only the content part of the log buffer is compressed, without considering the filler entry.

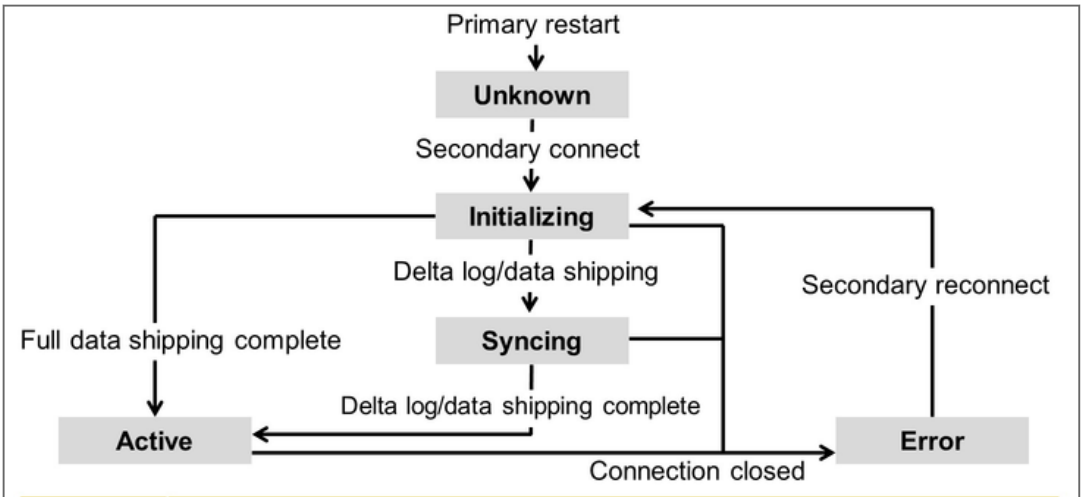
The activation of the compression reduces the required network bandwidth, but at the same time there is some CPU overhead for compressing and decompressing the information. Using compression is particularly useful in the case of long distances between primary and secondary sites or in the case of bandwidth limitations.

Checking and Monitoring of SAP HANA System Replication

After setting up the secondary system for system replication, you can monitor the status of the replication between the primary and the secondary system using the following tools:

- SAP HANA Cockpit
- SAP HANA Studio
- hdbnsutil

The current status of system replication can be checked with all of these tools.



Status	Description
Unknown	Secondary did not connect to primary since last restart of the primary
Initializing	Initial data transfer in progress. In this state, the secondary is not usable at all.
Syncing	Secondary is syncing again (for example, after a temporary connection loss or restart of the secondary).
Active	Initialization or sync with primary is complete and secondary is continuously replicating. No data loss will occur in SYNC mode.
Error	Error occurred on the connection

Figure 77: System Replication Status

Monitoring System Replication with SAP HANA Cockpit

To monitor SAP HANA system replication, you can use the System Replication tile in the SAP HANA Cockpit.

If system replication is configured, the System Replication tile provides information about the type of landscape (2-tier or 3-tier), the replication mode between the primary and the tier-2 secondary, the operation mode, and the overall replication status.

The System Replication tile displays the following states at a glance:

- Not configured (meaning system replication is not configured)
- All services are active and in sync
- All services are active, but not yet in sync
- Errors in replication

To check the status of replication in detail, choose the System Replication tile. The System Replication overview screen displays a graphical representation of the system replication landscape, configuration, and status. At the top, the “chain” of systems with their replication modes is shown, containing further information about the sites and the network connections between them.

The System Replication screen provides the following information:

- The name and role of the system, as well as the selected operation mode.

For the operation modes logreplay and logreplay_readaccess , a retention time estimation is also displayed. This is an estimation of the time left before the primary system starts to overwrite the RetainedFree marked log segments, and a full data shipping becomes necessary to get the primary and secondary systems back in sync after a disconnect situation. The estimated log full time is an estimation of the time left before the primary system runs into a log full. The value shown in the header shows the situation into which the system could run first: log retention or log full.

If the SQL ports of the secondary system are open for read access.

The replication mode used between the systems.

The current average redo log shipping time and the average size of shipped redo log buffers.

This describes how long it took on average to send redo log buffers to the secondary site, based on measurements over the last 24 hours.

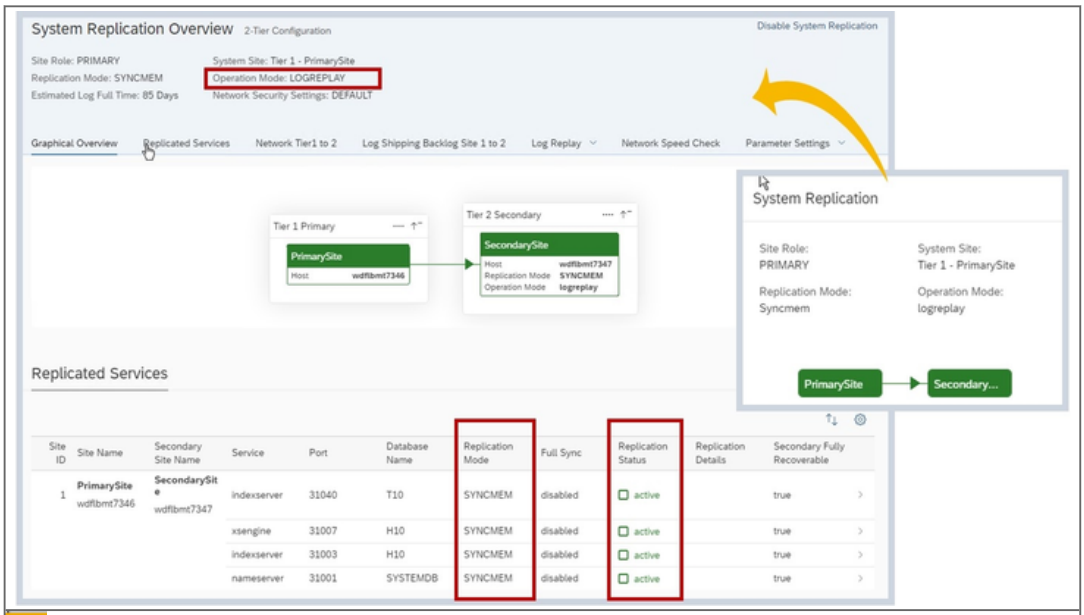


Figure 78: Check and Monitor SAP HANA System Replication

In addition, detailed information on system replication is provided in the tabs shown in the following figure.

Tab Name	Description
Replicated Services	The *Replicated Services* tab provides information on the replication status per site and service.
Network	The *Network* tab provides information on the time it took to ship the redo log to the secondary system and to write the redo log to the local log volume on disk. You can select the network connection that you want to analyze (for example, *Network Site 1 to 2* or *Network Site 2 to 3*). The graph dispalyed compares the local write wait time with the remote write wait time monitored over the last 24 hours.
Log Replay	The *Log Replay* tab provides a graphical representation on the delay of the secondary system. This tab is displayed if the chosen operation mode for the system replication landscape is `logreplay` or `logreplay_readaccess`. When this tab is activated for a secondary system, the log replay delay is shown for the last 24 hours. Furthermore, in this tab you can select to visualize the estimated log retention time as well as the estimated log full time for all system replication relevant services.
Network Speed Check	The *Network Speed Check* tab provides a way to measure the network speed of the system replication host-to-host network channel mappings.
Network Security Settings	The *Network Security Settings* tab displays the specific network security details configured between the primary and the secondary systems.

Figure 79: System Replication Overview Tabs

Selecting one row in the Replicated Services tab shows the details for the corresponding service grouped thematically, as in the following example for the index server. Because this information is context-sensitive, you only see the information required for this system. Therefore, because this example system is running in the logreplay operation mode, no information on delta data shipping is shown here. However, the context-sensitive information about the log replay delay is displayed. The delta between Last Log Position and Replayed Log Position indicates how far the log replay is behind on the secondary.

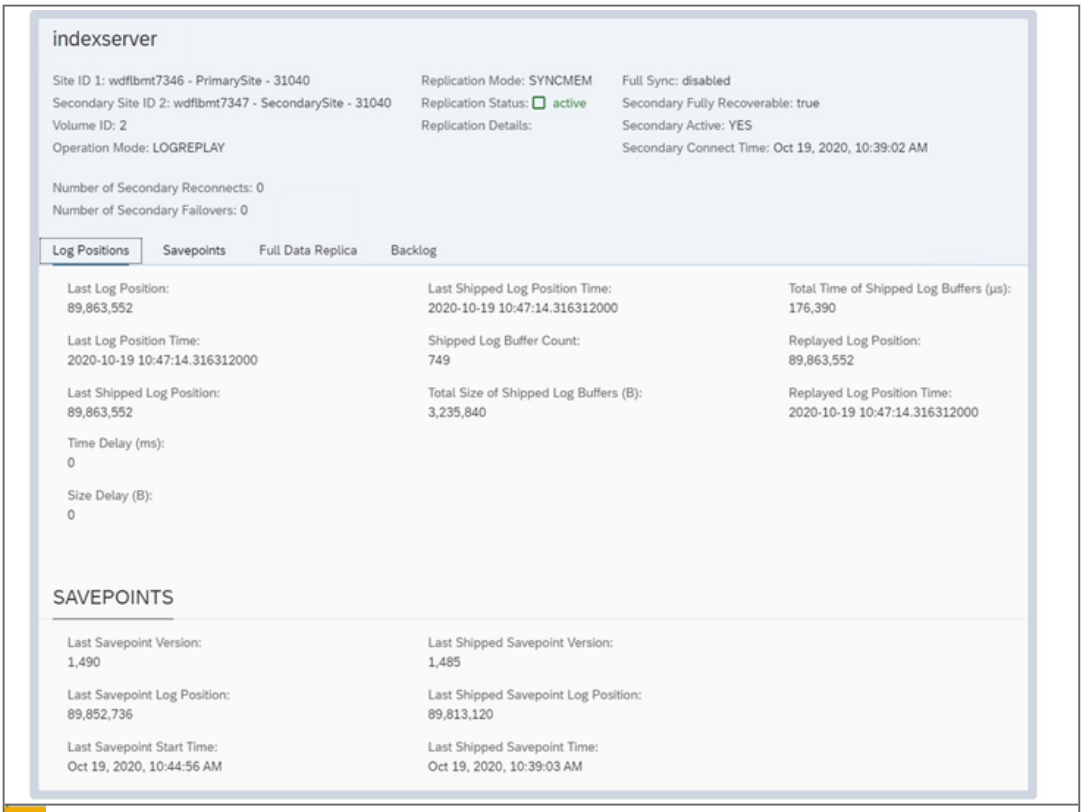


Figure 80: Details for the Status of a Specific Service

SAP HANA Cockpit for Secondary Management

The SAP HANA Cockpit distinguishes between a primary and a secondary system. On the SAP HANA Cockpit of the secondary system, the System Replication tile provides an initial overview of this site's state. From the System Replication Overview, you can initiate a takeover.

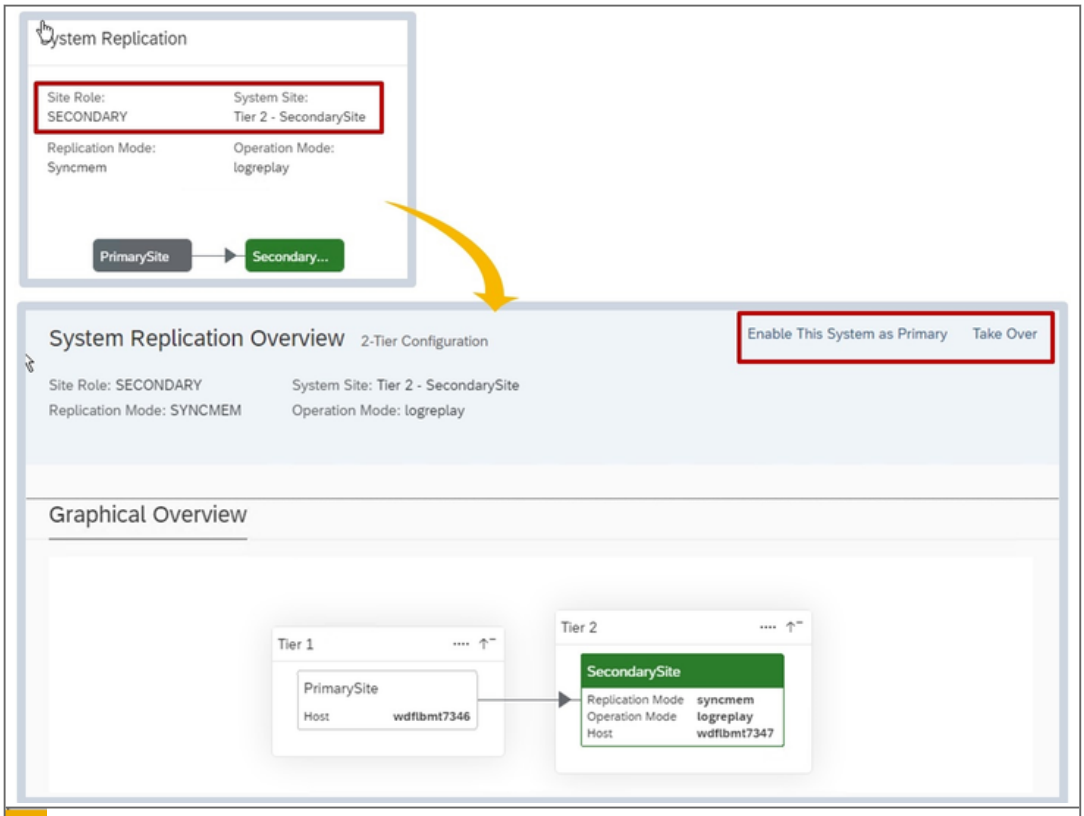


Figure 81: Management of Secondary Site

Monitoring System Replication with Command Line Tools and Scripts

Command Line Tools and Scripts to Monitor System Replication

hdbnsutil -sr_state

Checks if the primary and the secondary sites have been successfully enabled for system replication.

landscapeHostConfiguration.py

Checks the overall status of the primary system.

systemReplicationStatus.py

Checks the overall status of the system replication.

Note:
The Python scripts are located in the directory $DIR_INSTANCE/exe/python_support .

Command: hdbnsutil -sr_state

Primary Site:

```
h10adm@wdflbmt7346:/> hdbnsutil -sr_state
checking for active or inactive nameserver ...

System Replication State
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~
online: true

mode: primary
operation mode: primary
site id: 1
site name: PrimarySite

is source system: true
is secondary/consumer system: false
has secondaries/consumers attached: true
is a takeover active: false

Host Mappings:
~~~~~~~~~~~~~~

wdflbmt7346 -> [SecondarySite] wdflbmt7347
wdflbmt7346 -> [PrimarySite] wdflbmt7346


done.
```

Secondary Site:

```
h10adm@wdflbmt7347:/>  hdbnsutil -sr_state
checking for active or inactive nameserver ...

System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~
online: true

mode: syncmem
operation mode: logreplay
site id: 2
site name: SecondarySite

is source system: false
is secondary/consumer system: true
has secondaries/consumers attached: false
is a takeover active: false
active primary site: 1


Host Mappings:
~~~~~~~~~~~~~~

wdflbmt7347 -> [SecondarySite] wdflbmt7347
wdflbmt7347 -> [PrimarySite] wdflbmt7346

primary masters:wdflbmt7346

done.
```

Script: landscapeHostConfiguration.py

You can also gather information about the overall status of the sites and the system replication using Python scripts.

The landcapeHostConfiguration.py script shows the status of the primary system:

SAP HANA is OK.

SAP HANA will be OK after a host auto-failover, for example.

Not enough instances are started and a takeover would be useful.

Note:
The script does not tell you if the secondary system is ready for a takeover.

The script provides an overall status and a return code to match the overall host status.

A takeover is only recommended when the return code from the script is 1 (error).

Example:

```
<sid>adm># python $DIR_INSTANCE/exe/python_support/
landscapeHostConfiguration.py
| Host  | Host   | Host   |    ...    | NameServer | NameServer  | ...
|       | Active | Status |           | Config Role| Actual Role |
| ----- | ------ | ------ | --------- | ---------- | ----------- |
------
| host1 | yes    | ok     |    ...    | master 1   | master      | ...
| host2 | yes    | ok     |    ...    | master 2   | slave       | ...
overall host status: ok
```

The following host states are possible:

OK: System is OK.

WARNING: A host auto-failover to a standby host is taking place.

INFORMATION: The landscape is completely functional, but the current (actual) role of the host differs from the configured role.

ERROR: There are not enough active hosts.

Script: systemReplicationStatus.py

The systemReplicationStatus.py script shows the status of system replication.

Using systemReplicationStatus.py has the advantage of showing whether the secondary systems are in sync or not. This provides more confidence if a takeover is justified because if system replication was never in sync or is outdated, unexpected loss of data might occur.

Example:

```
h10adm@wdflbmt7346:/> python $DIR_INSTANCE/exe/python_support/
systemReplicationStatus.py
| Database | Host        | Service Name | Site Name   | Secondary   |
Secondary     | Replication |
|          |             |              |             | Host        |
Site Name     | Status      |
| -------- | ----------- | ------------ | ----------- | ----------- |
------------- | ----------- |
| SYSTEMDB | wdflbmt7346 | nameserver   | PrimarySite | wdflbmt7347 |
SecondarySite | ACTIVE      |
| H10      | wdflbmt7346 | xsengine     | PrimarySite | wdflbmt7347 |
SecondarySite | ACTIVE      |
| H10      | wdflbmt7346 | indexserver  | PrimarySite | wdflbmt7347 |
SecondarySite | ACTIVE      |

status system replication site "2": ACTIVE
overall system replication status: ACTIVE

Local System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

mode: PRIMARY
```

```
site id: 1
site name: PrimarySite
```

The additional parameter systemReplicationStatus.py --localhost restricts the execution of the python script to the host on which is it executed.

The script provides the following return codes:

10: No System Replication

11: Error

12: Unknown

13: Initializing

14: Syncing

15: Active

Monitoring System Replication Using SQL Statements

You can also get system replication-specific information directly from system views.

System Views Providing Information About System Replication

M_SERVICE_REPLICATION

Collects the history of data and log replication every hour.

M_SYSTEM_REPLICATION

Provides general system replication-relevant information about the whole system.

Note:
A set of complex SQL statements is available in SAP Note: 1969700 - SQL Statement Collection for SAP HANA. The section Replication System Replication includes some system replication-relevant statements. The Overview script provides information about the system replication landscape and the replication state for each service.

Monitoring System Replication Alerts

Specific alerts are issued by the primary system to warn you of potential problems.

System Replication Alerts

System Replication Connection Closed (Alert ID 78)

System Replication Configuration Parameter Mismatch (Alert ID 79)

System Replication Logreplay Backlog (Alert ID 94)

System Replication Increased Log Shipping Backlog (Alert ID 104)

The Connection Closed and Configuration Parameter Mismatch alerts are raised when a system replication connection is closed, or when there is a system replication configuration parameter mismatch.

The Logreplay Backlog alert is raised when the system replication logreplay backlog is increased. In this case, logreplay is delayed on the secondary site, causing a longer takeover time.

To identify the reason for the increased system replication logreplay backlog, check the state of the services on the secondary system. To get more information, monitor the secondary site. Possible causes for the increased system replication logreplay backlog can be, for example, a slow or non-functioning log replay, or a non-running service on the secondary system.

The Increased Log Shipping Backlog alert is raised when the system replication log shipping backlog is increased. In this case, the log shipping to the secondary system is delayed or does not work properly causing data loss on the secondary system in the case where a takeover is executed.

To identify the reason for the increased system replication log shipping backlog, check the status of the secondary system. Possible causes for the increased system replication log shipping backlog can be a slow network performance, connection problems, or other internal issues (for example, in the sync or syncmem replication modes).

Monitoring INI File Parameter Changes

Database parameters should be the same in the primary and secondary systems and are checked automatically. The configuration parameter checker reports on any differences between primary, secondary, and tier 3 secondary systems. In such a case, the parameter checker generates an alert.

With parameter replication activated, any changes made on the primary are automatically replicated to the secondary sites. Without this parameter replication activated, changes should be manually duplicated on the other system.

Parameter replication is off by default. It can be enabled and disabled on the primary site by using:

[inifile_checker]/replicate = true | false

The parameter checker is on by default. It can be enabled and disabled on the primary site by using:

[inifile_checker]/enable = true | false

Some parameters may have different settings on the primary and the secondary sites on purpose. One example is the global_allocation_limit parameter, where the secondary is used for other systems. By adding these parameters to the exclusion list, you can exclude them from checking.

LESSON SUMMARY
You should now be able to:

- Set up SAP HANA system replication

Unit 3
Lesson 4

Creating Tenant Databases in a System Replication Scenario

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Create tenant databases in a system replication scenario

Overview of SAP HANA System Replication with Tenant Databases

The usual SAP HANA system replication principles apply for tenant database systems.

Before you begin preparing a replication strategy for a SAP HANA system, you should be aware of the following important points:

- SAP HANA 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.

  Thereafter, 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 is not operational after takeover and must be recovered with backup and recovery.

- 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 are 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.

- It is possible to copy or move tenant databases between SAP HANA 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.

- SAP HANA systems can only be replicated as the whole system.
- If a new tenant database is created in a configured SAP HANA system replication, it must be backed up to participate in the replication.
- If an active tenant database is stopped in a running SAP HANA system replication, it is stopped on the secondary site as well.

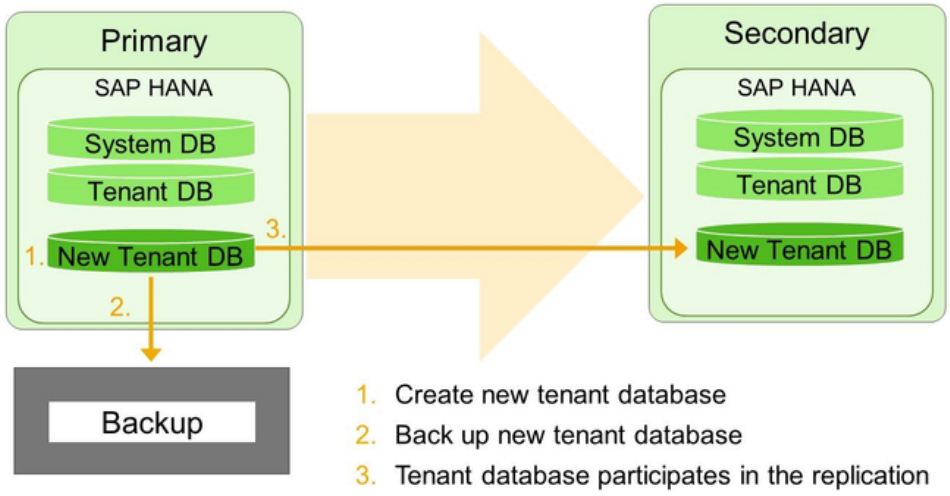


Figure 82: System Replication with Tenant Databases

LESSON SUMMARY
You should now be able to:

Create tenant databases in a system replication scenario

Unit 3
Lesson 5

Performing a Takeover on the Secondary System

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Perform a takeover on the secondary system

Perform Takeover

During a takeover, you switch your active system from the current primary system to the secondary system.

If your primary data center is not available, due to a disaster or for planned downtime for example, and a decision has been made to fail over to the secondary data center, you can perform a takeover on your secondary system.

In addition to the tools that may be used to monitor the overall system status when system replication is enabled, a script is provided with SAP HANA that helps you decide when a takeover should be performed.

We recommend that you use third-party, external tools to check if hosts, the network, and the data center are still available.

In addition, a script called landscapeHostConfiguration.py is provided so that SAP HANA itself can communicate the status of the primary system. It can communicate the following statuses:

- SAP HANA is OK.
- SAP HANA will be OK after a host auto-failover, for example.
- Not enough instances are started and a takeover would be useful.

A takeover is only recommended when the return code from the script is 1 (error).

Note:
The script does not tell you if the secondary system is ready for a takeover.

If a takeover occurs, the secondary site finds the latest savepoint in the data disk area. This is the starting point for a usual database restart, but many large data packages (main indexes) are preloaded in-memory, as on the primary data center before takeover. This supports the restart considerably. Based on this initial savepoint on the secondary data center, the log replay can start and roll the database forward to the latest point in time.

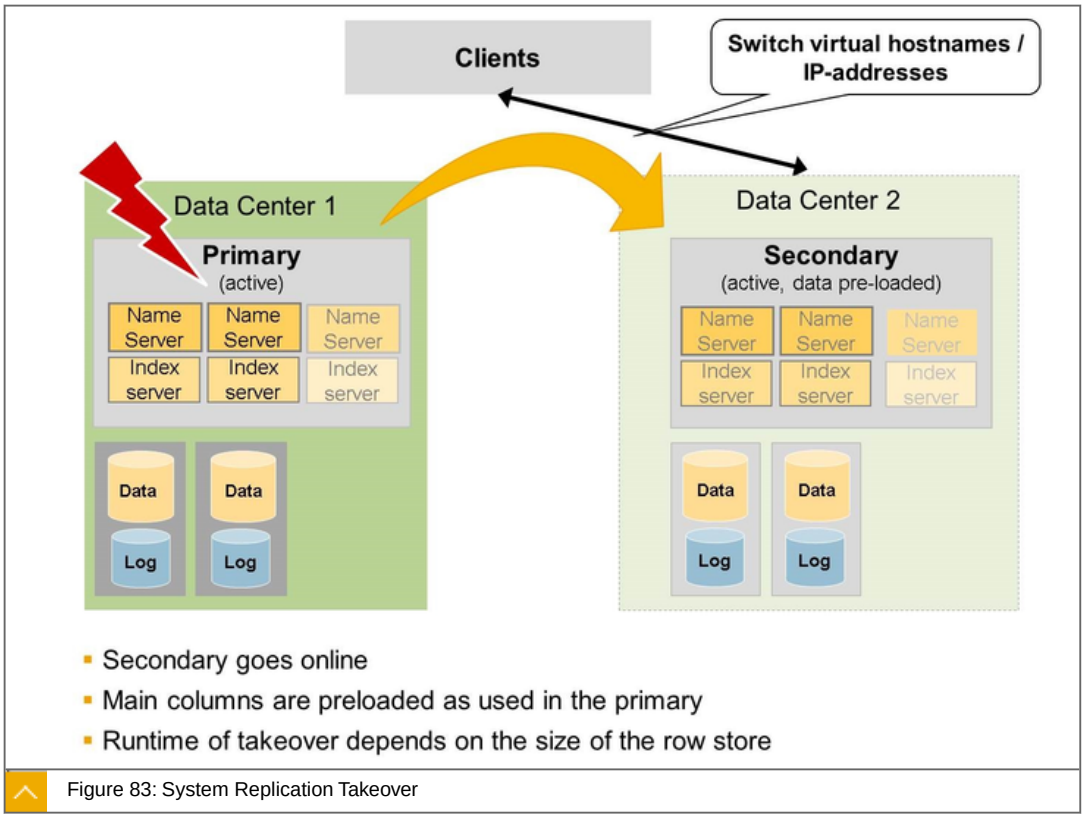


Figure 83: System Replication Takeover

The following decision guideline can help you decide if a takeover is advisable.

Takeover Decision Guideline

There are three main questions involved in deciding whether or not a takeover will improve the situation.

1. Can a takeover help at all?

   No: Do not perform a takeover.

   Yes: Proceed to question 2.

2. Can a takeover reduce the downtime duration?

   No: Do not perform a takeover.

   Yes: Proceed to question 3.

3. Can it be guaranteed that no data loss will result from the takeover?

   No: Evaluate the risk of data loss in the case of a takeover against that of data loss in case of no takeover, and against the impact of a longer downtime to bring back the primary site instead.

   Yes: Perform a takeover.

Note:
For more information on how to answer these questions, see SAP Note: 2063657.

You can use the getTakeoverRecommendation.py script to get takeover recommendations.

Takeover Recommendations are Given by the Script

getTakeoverRecommendation.py

Evaluates the status returned by the Python scripts:

- landscapeHostConfiguration.py
- systemReplicationStatus.py

These three possible states are returned:

- Takeover required
- Not decidable
- Possible

When the getTakeoverRecommendation script is called, it shows the takeover recommendation based on the current system state. However, when the primary system faces any error situation, the system replication status can no longer be determined. Therefore, the previous state should be saved and compared against the current state.

Example:

Primary Site:

This is a sample implementation of a python script that uses getTakeoverRecommendation to act as a minimalist cluster manager:

```
import time
import subprocess
from getTakeoverRecommendation import TakeoverDecision
def main():
   wasInSync = False
   while True:
      recommendation =
subprocess.call(["python","getTakeoverRecommendation.py","--
sapcontrol=1"])
      if not wasInSync and recommendation is
TakeoverDecision.Required:
         print "Primary defect & no sync => NO TAKEOVER"
      if wasInSync and recommendation is TakeoverDecision.Required:
         print "Primary defect & sync => TAKEOVER"
      nowInSync = recommendation is TakeoverDecision.Possible
      wasInSync = nowInSync
```

The output depends on the previous state with the result of the current call of getTakeoverRecommendation . If no sync state is reached, a takeover is not advised. But once the systems are in sync, the next error of the primary system will suggest a takeover. Any subsequent negative return value will reset the sync state, as it is no longer ensured that the replicated data is current.

Tools for Performing a Takeover

The takeover can be triggered using the following tools:

The SAP HANA Cockpit

SAP HANA Studio

hdbnsutil

The following steps are performed:

1. Trigger a takeover to the secondary system in the event of a disaster.
2. Register the former primary system as a new secondary when it becomes available again.

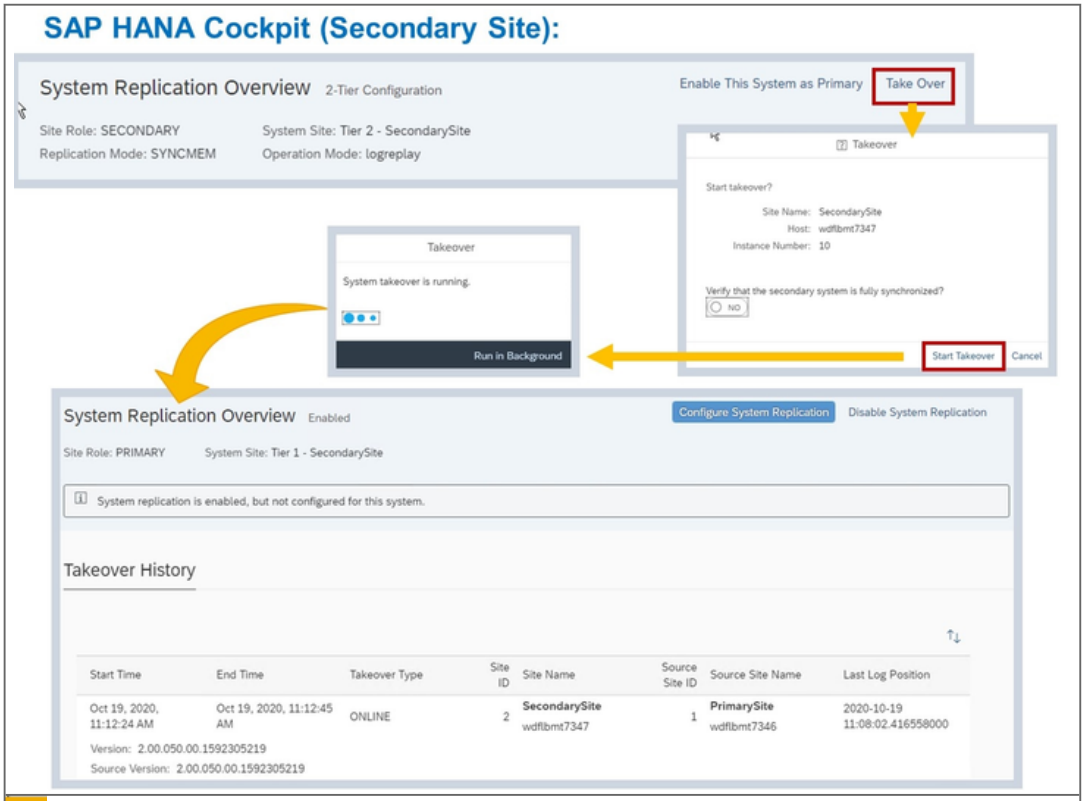


Figure 84: System Replication Takeover

Command Line Tool hdbnsutil

1. Perform a takeover on the secondary site:

```
hdbnsutil –sr_takeover
```

2. When the former primary site is available again it can be registered as the new secondary site:

```
hdbnsutil -sr_register --remoteHost=<new primary hostname>
--remoteInstance=<instance number>
--replicationMode=<sync/syncmem/async>
--operationMode=<delta_datashipping|logreplay>
--name=<siteName>
```

Note:
External cluster management software can be used to perform the client reconnect after takeover. Some of SAP's hardware partners offer an integration of SAP HANA high availability in their cluster management solutions.

Client Connection Recovery

To perform the takeover only on the SAP HANA system in most cases is not enough. Somehow, the client or application server needs to be able to continuously reach the SAP HANA system, no matter which site is currently the primary.

Methods for Client Connection Recovery

IP redirection

A virtual IP address is assigned to the virtual host name. In the case of a takeover, the virtual IP unbinds from the network adapter of the primary system and binds to the network adapter of the secondary system.

DNS redirection

In this scenario, the IP for the host name in the DNS is changed from the address of the primary system to the address of the secondary system.

Both methods have their advantages, but the method is mostly decided by IT policies and the existing configuration. If there are no existing constraints, IP redirection has the clear benefit of being faster to process in a script rather than synchronizing changes of DNS entries over a global network.

SAP HANA offers the so-called ”HA/DR providers” that are capable of informing external entities about activities inside SAP HANA scale-out (such as host auto-failover) and SAP HANA system replication setups. In a Python script, actions can be defined that should be executed before or after certain SAP HANA activities, such as startup, shutdown, failover, takeover, connection change, and so on. One example of these HA/DR providers, or “hooks”, is moving virtual IP addresses after a takeover in SAP HANA system replication.

Additionally, external cluster management software can be used to perform the client reconnect after takeover.

Takeover History

Monitoring View Providing Information About Takeover History

M_SYSTEM_REPLICATION_TAKEOVER_HISTORY

The M_SYSTEM_REPLICATION_TAKEOVER_HISTORY monitoring view provides information about take-overs in SAP HANA system replication (HSR) and when HSR was activated or reactivated.

During take-over, the content of the view is also moved to the system taking over, so that the complete take-over history is available.

Information provided by system view M_SYSTEM_REPLICATION_TAKEOVER_HISTORY
Execution end time for takeover of the transaction domain
Execution start time for takeover of the transaction domain
Master log position, that has been reached by takeover
Time that has been reached by takeover
Master nameserver host at takeover time
Operation mode at takeover time
Replication mode at takeover time
Replication status at takeover time
Highest master log position, that has been shipped before executing takeover
Time of the last shipped log buffer before executing takeover
Logical name provided by the site administrator at takeover time
Generated ID of the secondary site at takeover time
Source site master nameserver host at takeover time
Logical name for the source site provided by the site administrator at takeover time
Generated ID of the source site at takeover time
Source site SAP HANA version
End time of the takeover command
Start time of the takeover command
Indicates how the system went online: ONLINE: online takeover, OFFLINE: offline takeover, TIMETRAVEL: after time travel
SAP HANA version for the site that is executing the takeover

Figure 85: Takeover History

Implementing Takeover Hooks

Takeover Hooks

Takeover hooks are provided by SAP HANA in the form of a Python script template.

Pre- and post-takeover actions are implemented in this script, which are then executed by the name server before or after the takeover.

Therefore, the SAP HANA name server provides a Python-based API that is called at important points of the host auto-failover and the system replication takeover process.

There are a number of pre-takeover, post-takeover, and general hooks available.

These so called “hooks” can be used for arbitrary operations that need to be executed. One of the most important uses of the failover hooks is moving around a virtual IP address (in conjunction with STONITH).

There are other purposes like starting tools and applications on certain hosts after failover, or even stopping DEV or QA SAP HANA instances on secondary sites before takeover. Multiple failover hooks can be installed and used in parallel with a defined execution order.

The failover hooks are included in SAP HANA. SAP HANA comes with its own Python interpreter, which is used for interpreting the user defined failover hooks. The failover hook API also has a version number.

You can adapt Python files delivered with SAP HANA to create your own HA/DR provider. This allows you to integrate, for example, SAP HANA failover mechanisms into your existing scripts.

To create your own HA/DR provider, use the HADRDummy.pyscript (located in the $DIR_SYSEXE/python_support/hdb_ha_dr directory) as a template for implementing SAP HANA failover mechanisms in your own scripts.

After implementation of the basic HA/DR provider, you can add the methods listed in the figure, Hook Methods, to your provider.

Name	Trigger
startup()	Beginning of nameserver's start up phase
shutdown()	Just before the nameserver exists
failover()	As soon as the nameserver made a decision about the new role
stonith()	As soon as the nameserver made the decision about the new role
preTakeover()	As soon as the hdbnsutil -sr_takeover command is issued
postTakeover()	As soon as all services with a volume return from their assign-call (open SQL port)
srConnectionChanged()	As soon as one of the replicating services loses or (re-)establishes the system replication connection
srServiceStateChanged()	As soon as the nameserver made a decision about the new state
srReadAccessInitialized()	As soon as a tenant database or the system database is ready to accept SQL read queries on a read enabled secondary system

Figure 86: Hook Methods

As an example, the srServiceStateChanged() HA/DR Provider Hook reports changed service states. It notices that an SAP HANA service is currently stopping or crashing. This knowledge can be used to reduce the takeover (detection) time, especially in systems with huge index servers.

Note:
The procedure for creating a HA/DR provider, and the available hook methods, are described in detail in the SAP HANA Administration Guide.

Take-over with Handshake

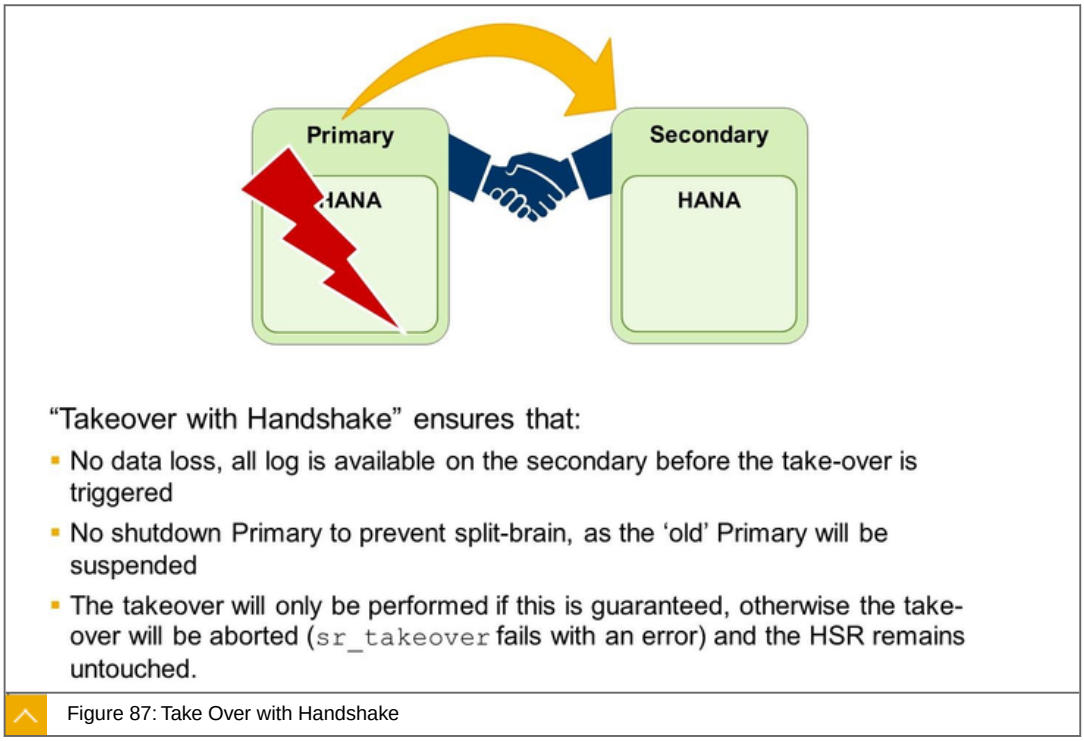


Figure 87: Take Over with Handshake

The takeover with handshake ensures that all of the sent redo log is written to disk on the secondary system.

During a planned takeover, it is important to ensure that no data gets lost (all primary updates must be available on the secondary system), and the former primary system is isolated to avoid a split-brain situation with multiple active primary systems.

The takeover with handshake is ideal for a safe planned takeover while the primary is still running. All new writing transactions on the primary system are suspended and the takeover is only executed when the redo log is available on the secondary system. When performing a takeover with handshake, it is not required to check the replication status or to stop the old primary before the takeover.

In a nutshell, a takeover with handshake avoids:

Data loss, because the log is available on the secondary system before the takeover is triggered.

Split-brain situations, because the former primary will be suspended.

Note:
The takeover with handshake will only be performed if the two previously mentioned conditions are guaranteed. Otherwise, the takeover will be aborted and the primary resumed.

You can trigger a takeover with handshake using hdbnsutil -sr_takeover -–suspendPrimary on the secondary system.

If a primary service cannot be accessed or a service replication is not active or in sync, the takeover will be aborted and reported as an error. In this case, there is no impact on the system and the replication remains as it was. The suspended primary can be unblocked using the -sr_register hdnsutil command.

Invisible Takeover

During an invisible takeover or a restart, the session's state needs to be recovered and restored to the new primary system.

During a standard takeover you switch your active system from the current primary system to the secondary system. After a standard takeover, the primary system loses all connections to the client. Moreover, the secondary system is not aware of the previous connections, which existed between the client and the primary system. This is different in an invisible takeover.

You can perform an invisible takeover to achieve an automatic recovery of your sessions after takeover to your new primary system. For dedicated client applications, this takeover is invisible. In contrast to a standard takeover, an invisible takeover ensures that the client reconnects to the primary system and the sessions are restored to the secondary system.

An invisible takeover has two functions:

- Keep the physical connections between the client and the primary and secondary systems
- Restore the sessions to the secondary system

This seamless recovery is possible also when restarting the system (for example, after a system crash).

The session's state needs to be recovered and restored to the new primary system in an invisible takeover scenario, or to the new system in a restart scenario. The cross-layer between the session and the client library makes the seamless recovery possible. This cross-layer feature called **transparent session recovery** recovers the current session's state and the physical connection.

Note:
As a first step, the focus is on read SQL, while write transactions (including database cursors) still need to be restarted after take-over (similar to classical databases).

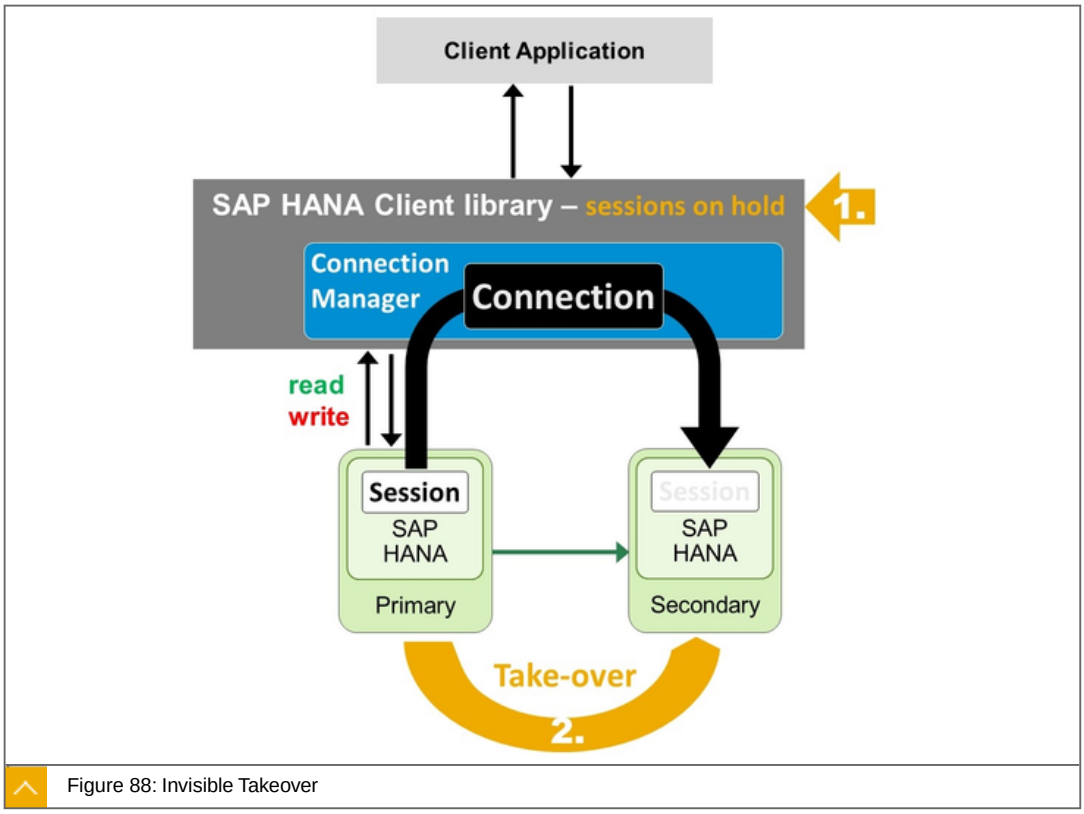


Figure 88: Invisible Takeover

Note:
The transparent session recovery is supported by SQLDBC for SAP HANA 2.0.

Up to SAP HANA 2.0 SPS 03, the implementation needs Active/Active system replication.

Configuration

The enable_session_recovery parameter controls the session recovery. The parameter is part of the indexserver.ini configuration file: indexserver.ini/session/ enable_session_recovery . The default value is **true** , recovering all session variables and restoring the client connections from the primary system to the secondary system. This parameter is configurable online, but the changes can be applied only to the connections established after making the changes.

Limitations

In SAP HANA 2.0 SPS04, almost all session variables from the current session context can be recovered, with the exception of the following limitations.

- Sessions that have created or updated a global temporary table with any DDL or DML commands will not be recovered. However, sessions which have created a local temporary table will be recovered without the table recovery.
- Only read transactions are supported. Ongoing write transactions will be rolled back with an error, and the session can be recovered when an application restarts the failed transaction with no explicit reconnect trial from the application.

Almost all session variables from the current session context are recovered.

When a response for a request is not successfully sent from the client to the server, the session is not recovered. However, sessions are still recovered when an SQL command is not sent from the client to the server.

LESSON SUMMARY
You should now be able to:

Perform a takeover on the secondary system

Unit 3
Lesson 6

Setting up Active/Active System Replication

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Set up Active/Active SAP HANA system replication

System Replication with a Read-Enabled Secondary Site

System replication with a read-enabled secondary site (Active/Active) enables read access on the secondary system.

Active/Active (read enabled) is integrated into the system replication solution and is activated with the logreplay_readaccess operation mode.

The logreplay_readaccess operation mode is similar to the logreplay operation mode with regard to the continuous log shipping, the redo log replay on the secondary system, and the required initial full data shipping and takeover.

> Caution:
> For this mode, the primary and secondary systems must have the same SAP HANA version. For this reason, read-only access to the secondary is not possible during a rolling upgrade until both versions are the same again.

Active/Active (read-enabled) is based on the continuous log replay feature. It inherits the following characteristics:

- Fast takeovers
- Reduced need for bandwidth in continuous operation
- Existing replication modes: SYNC (with or without the full sync option), SYNCMEM, ASYNC

Active/Active (read-enabled) offers integrated consistent views of data on the secondary site. These views can be delayed compared to the primary system. However, the secondary system identifies the exact delay. During an outage, all functions concentrate on the secondary system. As such, the sizing of the secondary system is important for the right performance in disaster scenarios. The following figure visualizes an Active/Active (read-enabled) system replication.

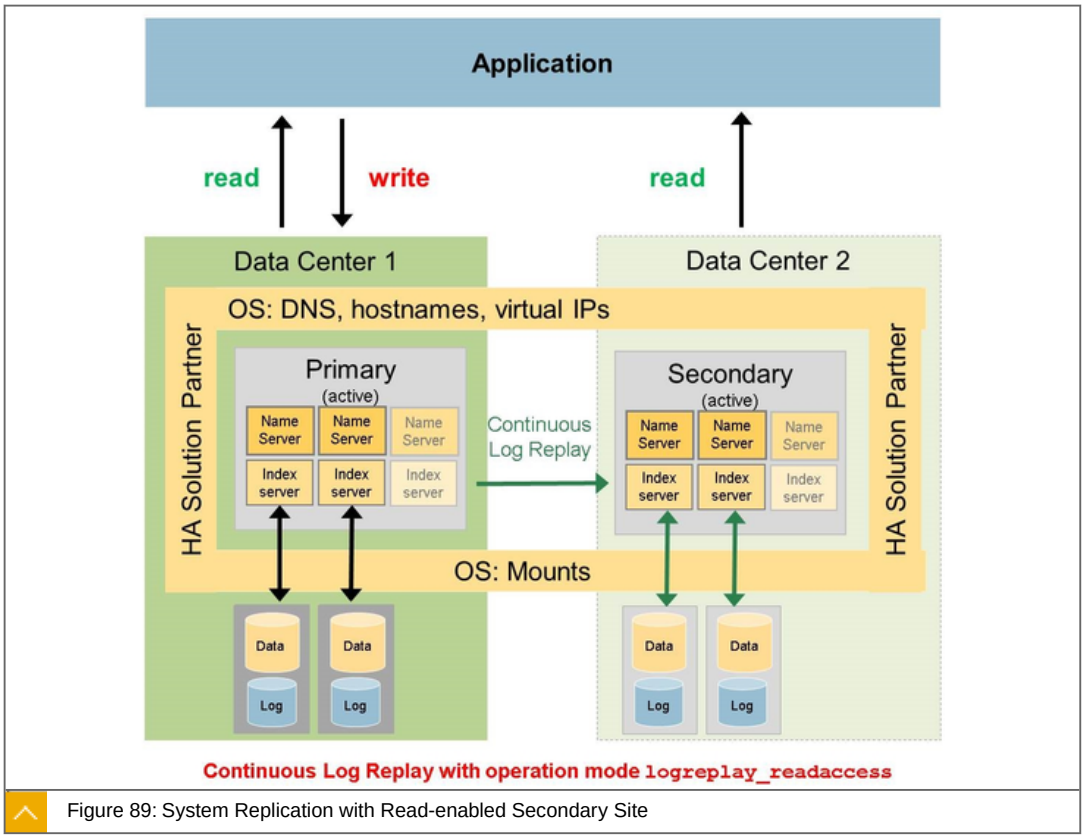


Figure 89: System Replication with Read-enabled Secondary Site

Note:
Active/Active (read-enabled) is only supported if the processors in the primary and secondary systems are both either Intel-based or IBM Power-based with the same byte ordering. A platform mixture is not supported.

Access Modes for Secondary Site

Connecting to an Active/Active (read-enabled) system allows you to take advantage of a secondary system for better overall performance.

There are two types of connections:

Explicit read-only connection to the secondary system

For this connection type, the application opens the connection to the secondary system. There is no session property sharing.

Implicit hint-based statement routing

Connections to the primary system can use hint-based routing statement execution to the secondary system on a per-statement basis.

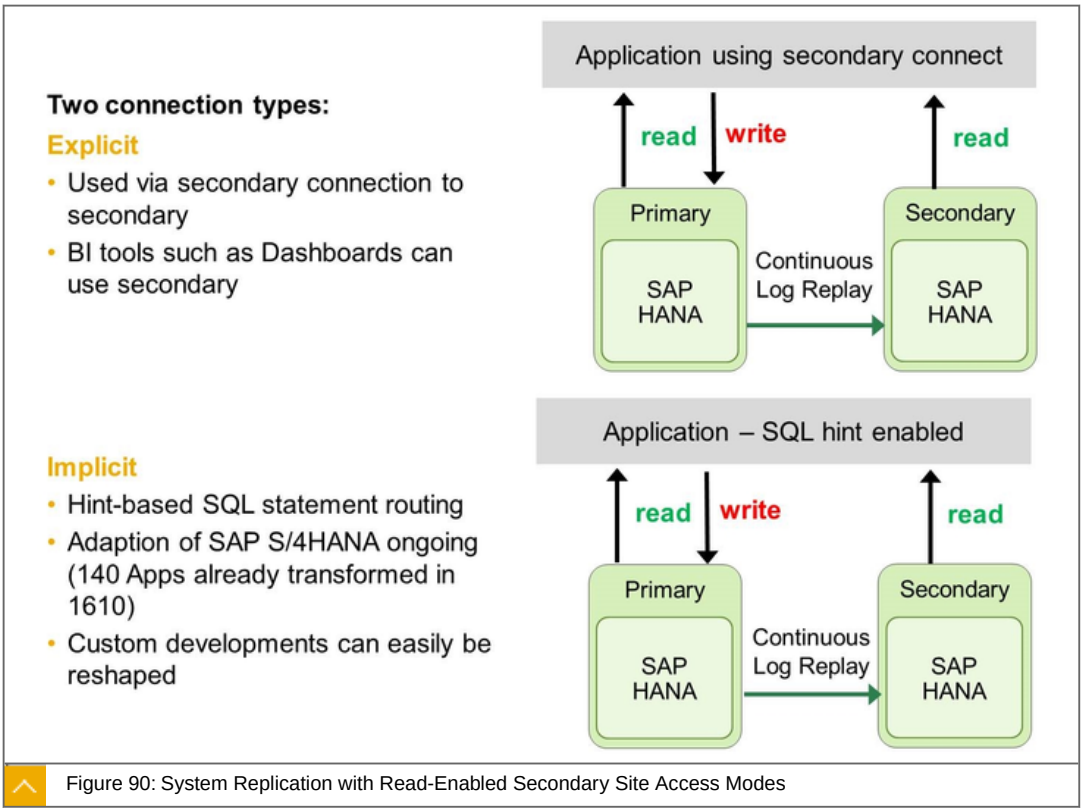


Figure 90: System Replication with Read-Enabled Secondary Site Access Modes

Using implicit hint-based statement routing, the SAP HANA client opens an additional connection to the secondary system according to the host information returned by the primary system.

This connection type unfolds as follows:

1. The SAP HANA client sends the statement-prepare with hint to the primary system.
2. The primary system decides where to execute the statement and returns the result to the SAP HANA client.
3. The SAP HANA client sends the statement execution call to the secondary system. Furthermore, the session property changes are delivered to the secondary system by the SAP HANA client. If the secondary system cannot execute the statement, it returns an error, and the SAP HANA client sends the statement to the primary system.

Memory Management Aspects and Support for Multiple SAP HANA Databases

When using Active/Active (read-enabled) system replication, several memory management aspects must be considered.

The total statement memory is limited to 50% of the global allocation limit, because 50% of the storage is reserved for log replay. Log replay should not fail because of storage limitations.

It is possible to use the read-enabled secondary for other SAP HANA systems, such as Development or QA environments. In this case, the following sizing conditions apply:

The secondary hardware must offer the same CPU and memory capacities as those offered by the primary, plus the resources for the additional system.

After a takeover, the system must be capable of handling both the primary's writing load and the secondary's reporting load.

LESSON SUMMARY
You should now be able to:

Set up Active/Active SAP HANA system replication

Unit 3
Lesson 7

Setting up SAP HANA System Replication with Secondary Time Travel

LESSON OBJECTIVES

After completing this lesson, you will be able to:

- Set up SAP HANA system replication with Secondary Time Travel

System Replication with Secondary Time Travel

SAP HANA system replication allows time travel for logical error mitigation. Therefore, you can start the secondary system in online mode from a previous point in time.

Secondary time travel can be used:

- To quickly access data that was deleted in the original system.
- To intentionally keep the secondary system's log replay delayed. This can be used to read older data from the secondary system, while the secondary keeps replicating.

To prepare the secondary system for time travel, snapshots are kept on the secondary system for a defined time travel period. These snapshots can be used later to start the system at an earlier point in time. An additional log is retained on the secondary system starting from the earliest time travel snapshot. After opening the old snapshot, the additional log has to be replayed to reach the requested point in time.

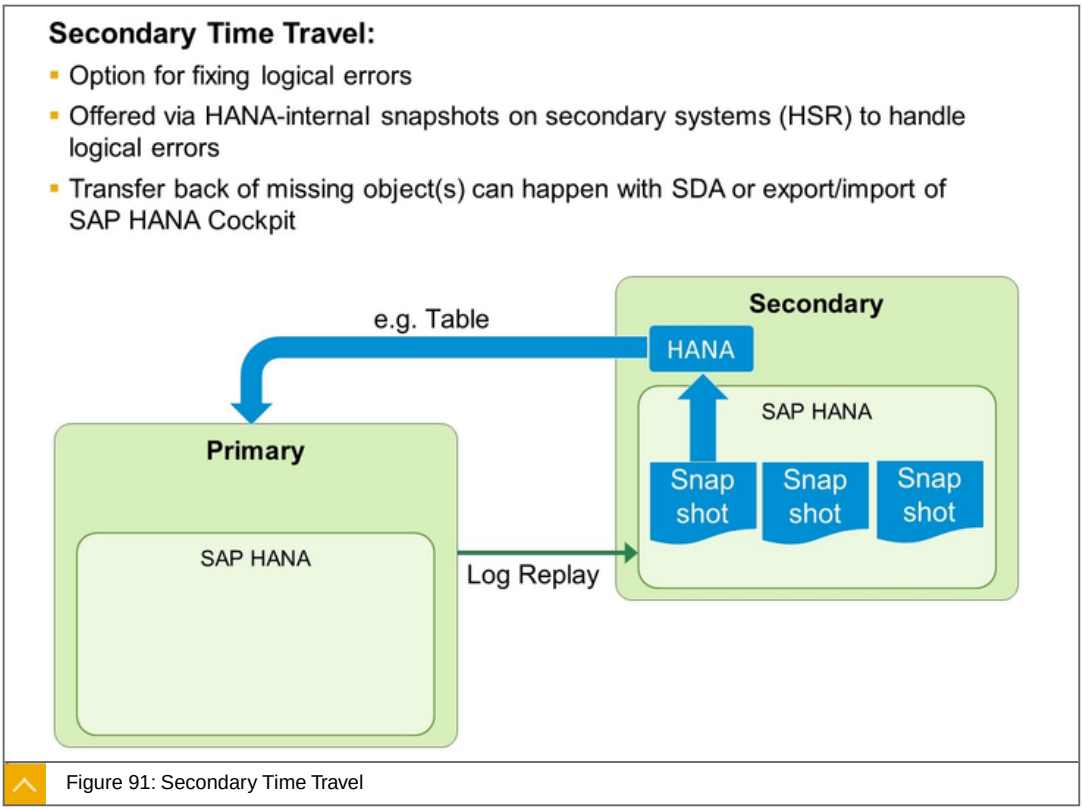


Figure 91: Secondary Time Travel

Note:
You can only use secondary time travel with the following operation modes: logreplay or logreplay_readaccess .

Configuration Parameters

Several parameters are available for configuring secondary time travel.

Use the following parameters to configure secondary time travel. The parameters are defined in the system_replication section of the global.ini file. All parameters are set on the secondary system.

Parameter	Unit	Def.	Description
timetravel_max_retention_time	Min.	0	If set to 0, secondary time travel is turned off. If this parameter is set to a value different from 0, the secondary system can be brought online up to the defined time period in the past.
timetravel_snapshot_creation_interval	Min.	1440 (24h)	Defines how frequently snapshots are created for secondary time travel. A new snapshot is created when the time period defined in this parameter has passed since the last snapshot creation. Snapshots older than the time period defined in time_travel_max_retention_time are dropped.
Parameter	**Values**	**Def.**	**Description**
timetravel_call_takeover_hooks	true, false	false	Indicates if takeover hooks should be called during secondary time travel. [Values: true, false]
timetravel_logreplay_mode	auto, manual	auto	Defines how the log replay is executed on the secondary system. [Values: auto, manual] ▪ Auto The log replay is done automatically up to the newest possible log position. ▪ Manual You must manually trigger the log replay up to the requested timestamp hdbnsutil command.

Figure 92: Secondary Time Travel – Parameter

Time travel snapshots are kept until they get older than the defined timetravel_max_retention_time parameter. If a takeover needs to be done from an earlier point in time, the snapshot that best fits the requested point in time is opened and the remaining changes are applied using logreplay.

Enable Secondary Time Travel

Enable Secondary Time Travel:

- Set the `timetravel_max_retention_time` parameter to define the time period in the past to which the secondary system can be brought back to.
- Set the `timetravel_snapshot_creation_interval` parameter to adjust the secondary snapshot creation (optional).

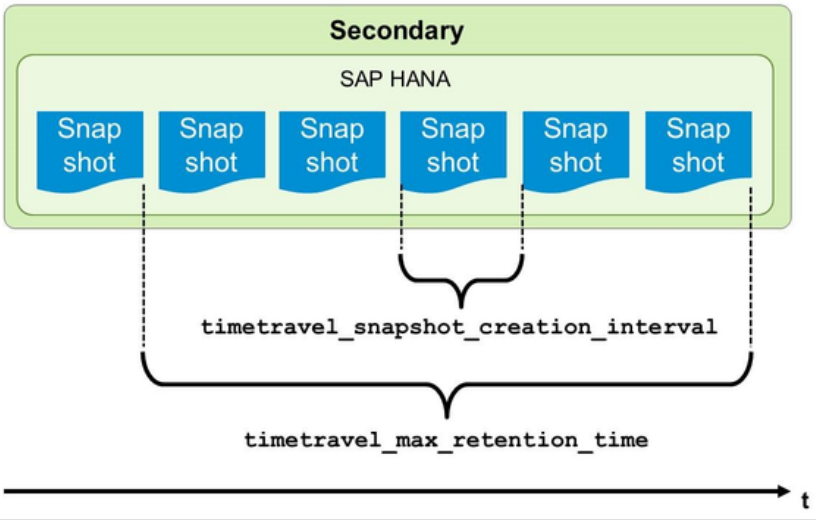


Figure 93: Enable Secondary Time Travel

You can edit the secondary .ini file directly and activate using: hdbnsutil -reconfig

Note:
Set the parameters carefully to avoid log full or disk full situations. For time travel to work, log and snapshots are kept online in the data area. Because of this, log and data grows on the secondary system when time travel is turned on. The system workload determines how much data is needed.

Note:
Topology changes are not considered for secondary time travel, and travel operation cannot be supported beyond these changes.

After setting the parameters, the secondary system begins retaining log information and keeping created snapshots. After retaining sufficient log information and data, the secondary system is ready for time travel.

Monitoring Secondary Time Travel

You can monitor the retaining log and the created snapshots.

To monitor secondary time travel, the secondary system must be online. The current time travel range cannot be determined when the secondary system is offline.

Monitoring Secondary Time Travel

Determine the valid range for which time travel can be executed:

- Monitor available snapshots using _SYS_DATABASES_SR_SITE_<sitename>.M_SNAPSHOTS at the primary site.
- The hdbnsutil -sr_timetravel --printRange command provides a range for each service in which time travel can be executed.
- Use SQL on the primary system with the _SYS_DATABASES_SR_SITE_<sitename>.M_SYSTEM_REPLICATION_TIMETRAVEL secondary proxy view.

Monitor the start time or log position of the system using: _SYS_DATABASES_SR_SITE_<sitename>.M_SYSTEM_REPLICATION_TAKEOVER_HISTORY

Execute Secondary Time Travel

You can start the secondary system in online mode from a previous point in time using the hdbnsutil -sr_timetravel command. During execution of hdbnsutil -sr_timetravel , the specified time and location are stored internally in a dedicated file in the SAP HANA directory. When calling hdbnsutil -sr_timetravel , it can be specified whether takeover hooks should be called. If the parameter is not explicitly specified, the default value from the configuration parameter timetravel_call_takeover_hooks is used.

The secondary system enters online mode at the specified point in time during restart. After restart, the other services read the requested point in time and open their persistence using this information. If the requested point in time cannot be reached, then time travel is aborted.

A check ensures that there are time travel snapshots older than the start time for each service.

Execute Secondary Time Travel

1. Stop the secondary SAP HANA system.
2. Execute:

   hdbnsutil -sr_timetravel --startTime=<startTime> [--callTakeoverHooks=on|off] [--comment="Your Comment"] .
3. Start the secondary SAP HANA system.

For startTime , use the following format specified in UTC: dd.mm.yyyy - hh.mm.ss

You can specify whether takeover hooks should be called. If the timetravel_call_takeover_hooks parameter is not explicitly specified, takeover hooks will not be called.

Use the --comment option to add a reason for the time travel. This comment is displayed in the M_SYSTEM_REPLICATION_TAKEOVER_HISTORY monitoring view in the COMMENTS column.

The secondary system will enter online mode at the specified point in time during restart. After restart, the other services read the requested point in time and open their persistence using this information. If the requested point in time cannot be reached, then time travel will be aborted. A check ensures that there are time travel snapshots older than the start time for each service.

Note:

The call of hdbnsutil -sr_timetravel on the secondary system is not yet offered in the system replication app of the SAP HANA cockpit.

Execute Secondary Time Travel While Replication Continues

You can start the log replay at a previous point in time to read older data from the secondary system, while the secondary keeps replicating.

Execute Secondary Time Travel While Replication Continues

1. Stop the secondary SAP HANA system.
2. Execute:

   hdbnsutil -sr_timetravel --startTime=<startTime> --startMode=replicate
3. Start the secondary SAP HANA system.
4. Optional: Trigger the log replay manually with:

   hdbnsutil -sr_recoveruntil {--endTime=<timestamp> |max} [--nowait]
5. Optional: Stop the manual replay mode by setting the timetravel_logreplay_mode parameter back to auto or using:

hdbnsutil -sr_replaymode --mode={auto|manual}

After the system has started, the persistence has been opened on the specified point in time (--startTime). It is replicating the log, and log replay is not running.

In this state, log replay can be executed manually. By executing multiple replay until calls, the secondary site can be step wise rolled forward and data can be accessed in read access mode after each replay step.

Use max to trigger the log replay up to the newest possible point in time. In this case, the target timestamp is automatically determined by checking the valid time travel range for each service.

Use the --nowait option to specify if the command should be executed asynchronously.

Setup for the Exercises

In the following exercise, you set up the SAP HANA system replication with secondary time travel.

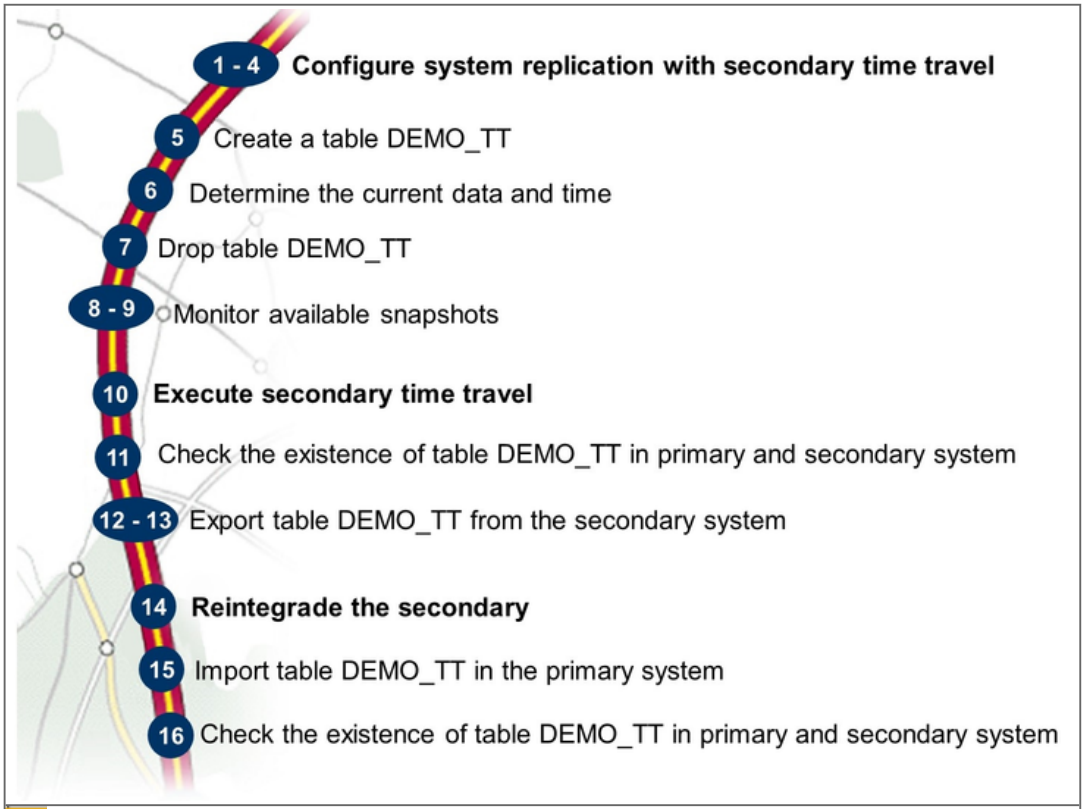


Figure 94: Secondary Time Travel Exercise Overview

LESSON SUMMARY
You should now be able to:

Set up SAP HANA system replication with Secondary Time Travel

Unit 3
Lesson 8

Explaining Zero Downtime Maintenance

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Explain Zero Downtime Maintenance

Zero Downtime Maintenance

You can use SAP HANA system replication to upgrade your SAP HANA systems, as the secondary system can run with a higher software version than the primary system.

Near Zero Downtime Upgrade

SAP HANA offers zero downtime maintenance, together with SAP HANA system replication. You can use system replication to upgrade your SAP HANA systems because the secondary system can run with a higher software version than the primary system.

As a prerequisite, system replication must be configured and active between two identical SAP HANA systems.

If system replication is active, you can first upgrade the secondary system to a new revision so that it can take over the role of the primary system. The takeover only requires a few minutes and the committed transactions or data are not lost. You can then perform an upgrade on the primary system, which is now in the role of secondary.

The secondary system can be initially installed with the new software version, or upgraded to the new software version when replication is already configured. After you upgrade the secondary, you must replicate all data to the secondary system, which already has the new software version. When the secondary system is ACTIVE (all services have synced), you can execute a takeover on the secondary system. This step makes the secondary system productive with the new software version.

In an Active/Active (read enabled) system replication setup, the version of the primary and the secondary systems must be identical. For the near zero downtime upgrade to work, the operation mode on the secondary system is automatically set to logreplay . Like this, the two systems can get back in sync before the takeover step. To reestablish the Active/Active (read enabled) landscape at the end, the logreplay_readaccess operation mode must be explicitly specified during the former registration of the primary system as a new secondary system.

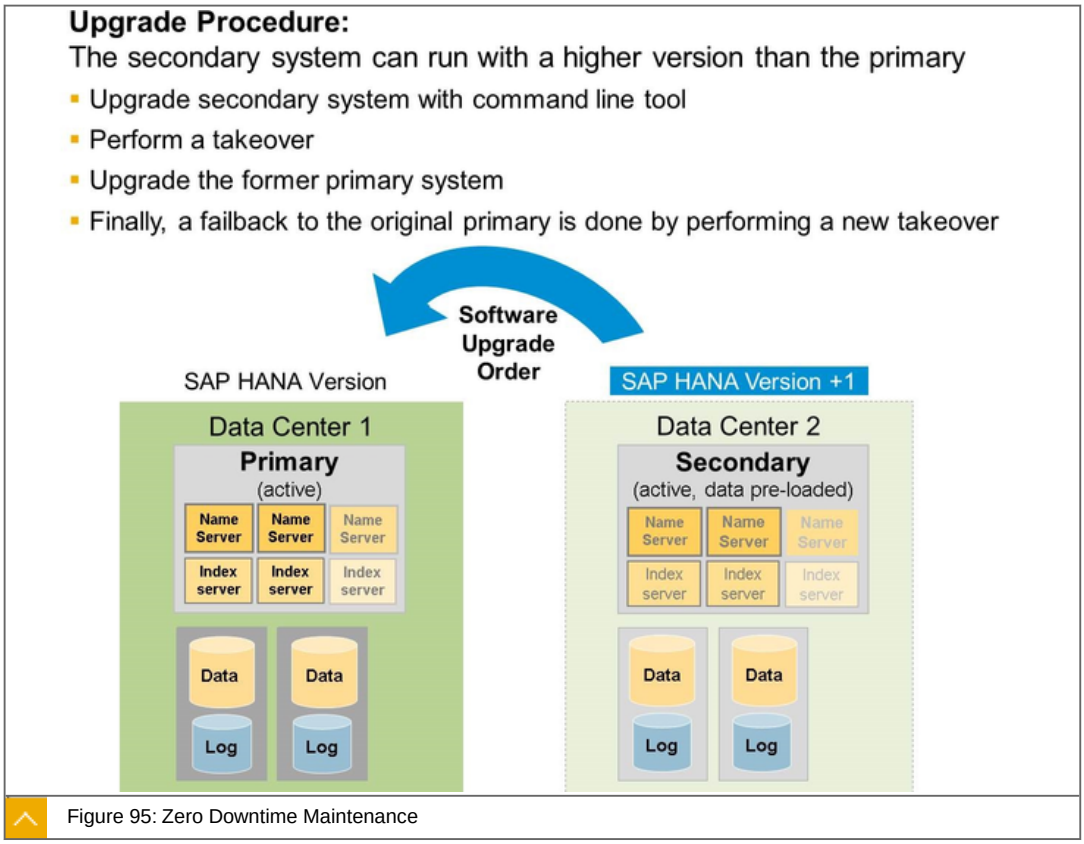


Figure 95: Zero Downtime Maintenance

Prerequisites

You configured a user in the local userstore under the SRTAKEOVER key.

Therefore, create a <myrepouser> user with the necessary privileges to import the repository content at takeover time on the primary and secondary systems.

Set the user store entry under the SRTAKEOVER key using the following command:

```
hdbuserstore SET SRTAKEOVER <public hostname>:<sqlport>
<myrepouser> <myrepousers_password>
```

System replication is configured and active between two identical SAP HANA systems:

The primary system is the production system.

The secondary system will become the production system after the upgrade.

The prerequisite is to run both systems with the same endianness.

The process, which is described in detail in the SAP HANA Administration Guide, looks like the following:

1. Upgrade the secondary system:

   ./hdblcm --action=update

2. Verify that system replication is active and that all services are in sync.
3. Stop the primary system.
4. Perform a takeover on the secondary system, including switching virtual IP addresses to the secondary system, and start using it productively:

hdbnsutil -sr_takeover

5. Upgrade the previous primary system without starting the system. This is necessary because otherwise the primary has to be stopped again before it can be registered as the secondary.

   ./ hdblcm --action=update --hdbupd_server_nostart

6. Register the previous primary system as the secondary system:

   ./hdbnsutil –sr_register …

7. Start the previous primary system as the secondary system.

Zero Downtime Maintenance Featured by SAP NetWeaver ABAP Stack

To achieve a real zero downtime upgrade from the application server perspective, see SAP Note: 1913302 — Connectivity suspend of Appserver during takeover.

Based on the connectivity suspend feature of the SAP NetWeaver ABAP stack, DBSL of the database interface decouples transaction management between ABAP and the SAP HANA database. This keeps the transaction on the ABAP layer alive and allows the change of components (software versions) on the layers below the secondary (shadow) SAP HANA instance.

Hardware Exchange

Additionally, as described in SAP Note: 1984882 — Using HANA system replication for Hardware Exchange with Minimum Downtime, hardware can be exchanged with minimal downtime using SAP HANA system replication.

LESSON SUMMARY
You should now be able to:

Explain Zero Downtime Maintenance

Unit 3
Lesson 9

Introducing Multitier and Multitarget System Replication

LESSON OBJECTIVES

After completing this lesson, you will be able to:

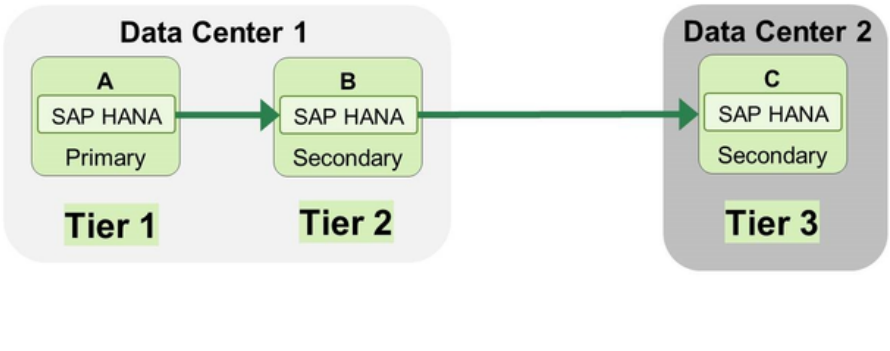
- Explain Multitier and Multitarget System Replication

Multitier System Replication

To offer higher levels of availability, you can link together multiple systems in a SAP HANA multitier system replication landscape.

With Multitier system replication, a tier 2 system replication setup can be used as the source for replication in a chained setup of primary site, tier 2 secondary site, and tier 3 secondary site.

After setting up a basic system replication scenario, you add a third system to provide another level of redundancy. In a multitier setup, the primary system is always on tier 1, a tier 2 secondary has a primary system as its replication source, and a tier 3 secondary has the tier 2 secondary as its replication source.



Allowed replication mode configurations:

Tier 1 to Tier 2	Tier 2 to Tier3
SYNC	SYNC
	SYNCMEM
	ASYNC
SYNCMEM	SYNC
	SYNCMEM
	ASYNC
ASYNC	ASYNC

Figure 96: Multitier System Replication: Overview

In general, multitier system replication does not allow operation mode mixtures. However, there is one exception. If the logreplay_readaccess operation mode is configured between tier 1 and tier 2, the logreplay operation mode can be configured between tier 2 and tier 3.

Set Up SAP HANA Multitier System Replication

You can configure multitier system replication using the following tools:

The SAP HANA Cockpit

hdbnsutil

SAP HANA Studio

To set up multitier system replication, you have installed and configured three identical, independently operational SAP HANA systems: a primary system, a tier 2 secondary system, and a tier 3 secondary system.

To use hdbnsutil to set up multitier system replication, you must perform the following steps. In this scenario, there are three SAP HANA systems: **A**, **B**, and **C**, named Tier1 , Tier2, and Tier3 respectively. In addition, in this scenario, multitier system replication supports a tier 2 secondary with sync replication mode and a tier 3 secondary with async replication mode.

Set up an SAP HANA Multitier System Replication Using hdbnsutil

1. Start SAP HANA system **A** and back up the system database and all tenant databases.
2. Enable system replication and give system **A** a logical name:

   hdbnsutil -sr_enable --name=Tier1
3. Stop the tier 2 secondary system **B**.
4. Register system **B** as tier 2:

```
hdbnsutil -sr_register --replicationMode=sync --name=Tier2
      --remoteInstance=<instID> --remoteHost=<hostA>
```

5. Start the tier 2 secondary system **B**.
6. Enable tier 2 as the source for a tier 3 secondary system:

   hdbnsutil -sr_enable
7. Stop the tier 3 secondary system **C**.
8. Register system **C** as tier 3:

```
hdbnsutil -sr_register --replicationMode=async --name=Tier3
      --remoteInstance=<instID> --remoteHost=<hostB>
```

9. Start the tier 3 secondary system **C**.

Using the SAP HANA Cockpit, you can set up multitier system replication in one step. After you have entered the system details for tier 1 and tier 2, you can add the details for the tier 3 system.

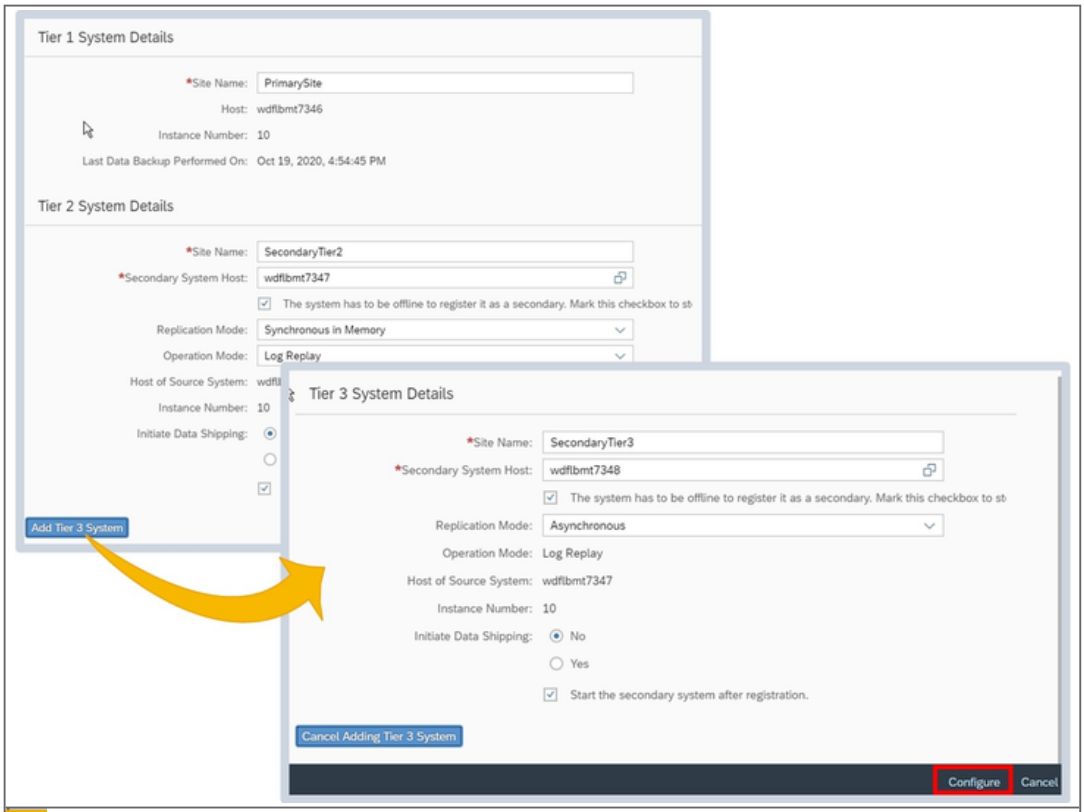


Figure 97: Set up Multitier System Replication

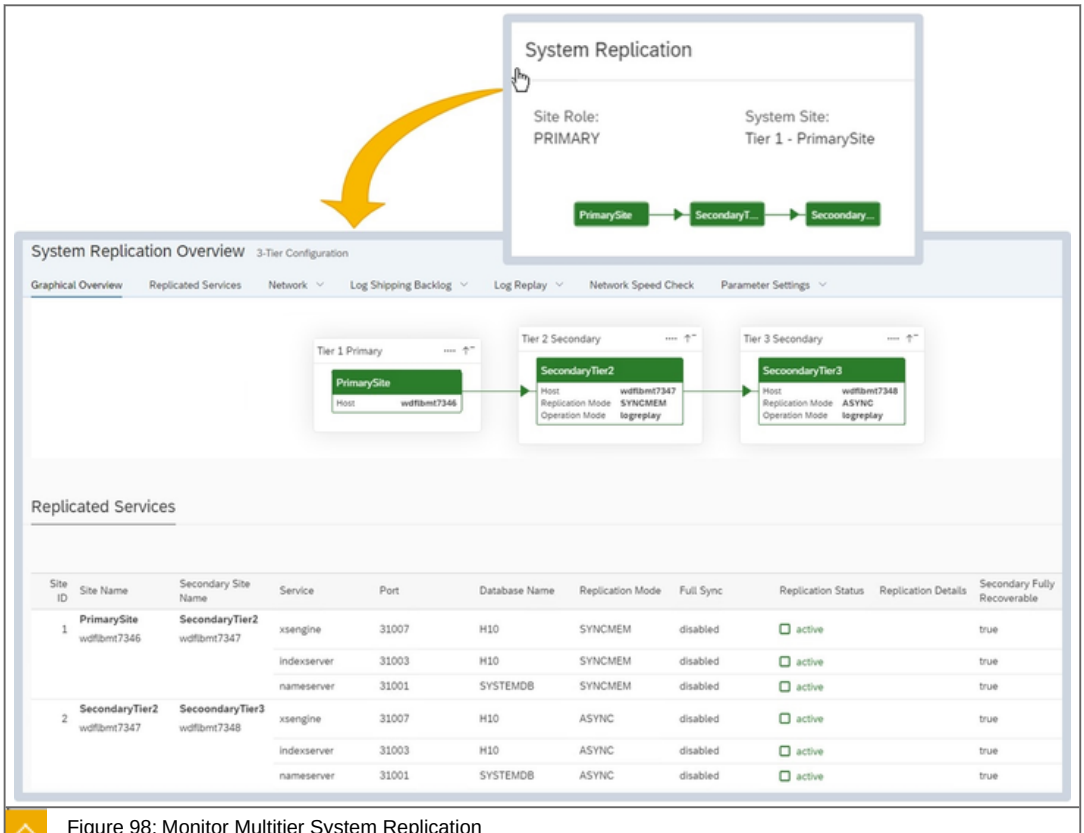


Figure 98: Monitor Multitier System Replication

If the primary system fails, a takeover to the tier 2 secondary system is done. Once your failed site is operational again, you can attach it as a tier 3 secondary system or you can restore the original multitier system replication configuration.

Multitarget System Replication

In a multitarget system replication, the primary system can replicate data changes to more than one secondary system.

Multitarget system replication can bring advantages for several use cases:

- Update scenarios
- Rearrangements of system replication multitier chains
- Reaching higher availability (before stopping existing structures, new structures can be built and established)

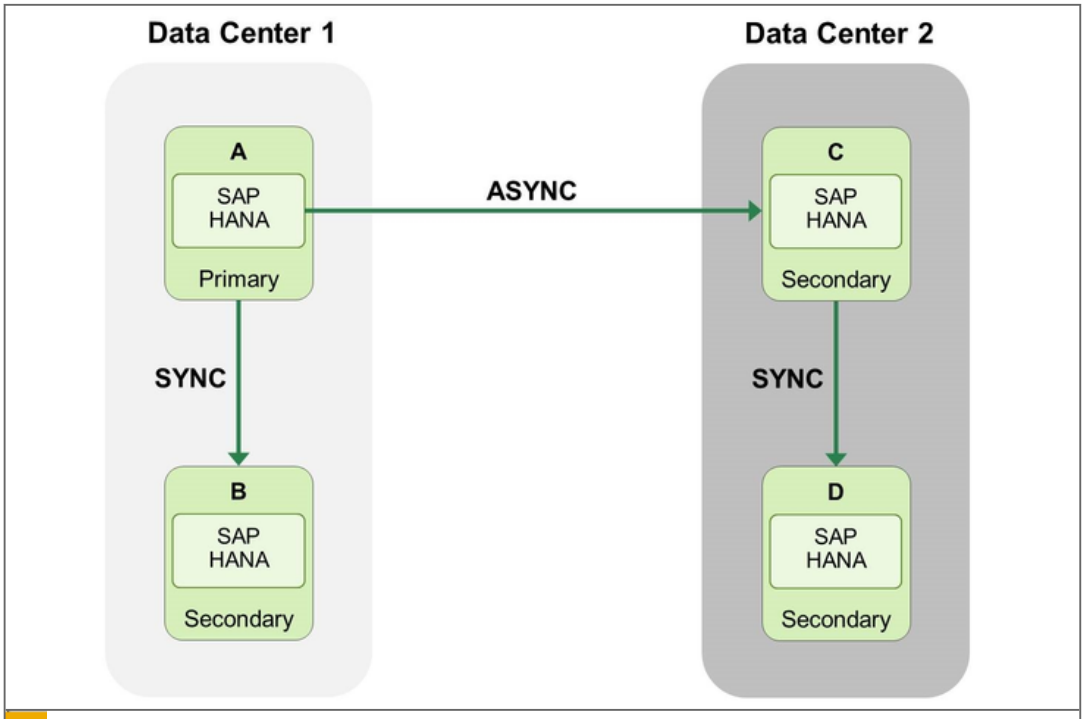


Figure 99: Multitarget Replication – Overview

Primary system A in data center 1 replicates data changes to secondary system B in the same data center. Primary system A also replicates data changes to secondary system C in data center 2. Secondary system C is a source system for a further secondary system D located in the same data center with system C.

In a multitarget system replication, the secondary systems can be configured to automatically re-register to a new source system when the original source system is unavailable.

In a multitarget system replication setup, you can configure multiple secondary systems as Active/Active (read enabled). Only one of these secondary systems can be accessed using hint-based statement routing; the others must be accessed using a direct connection.

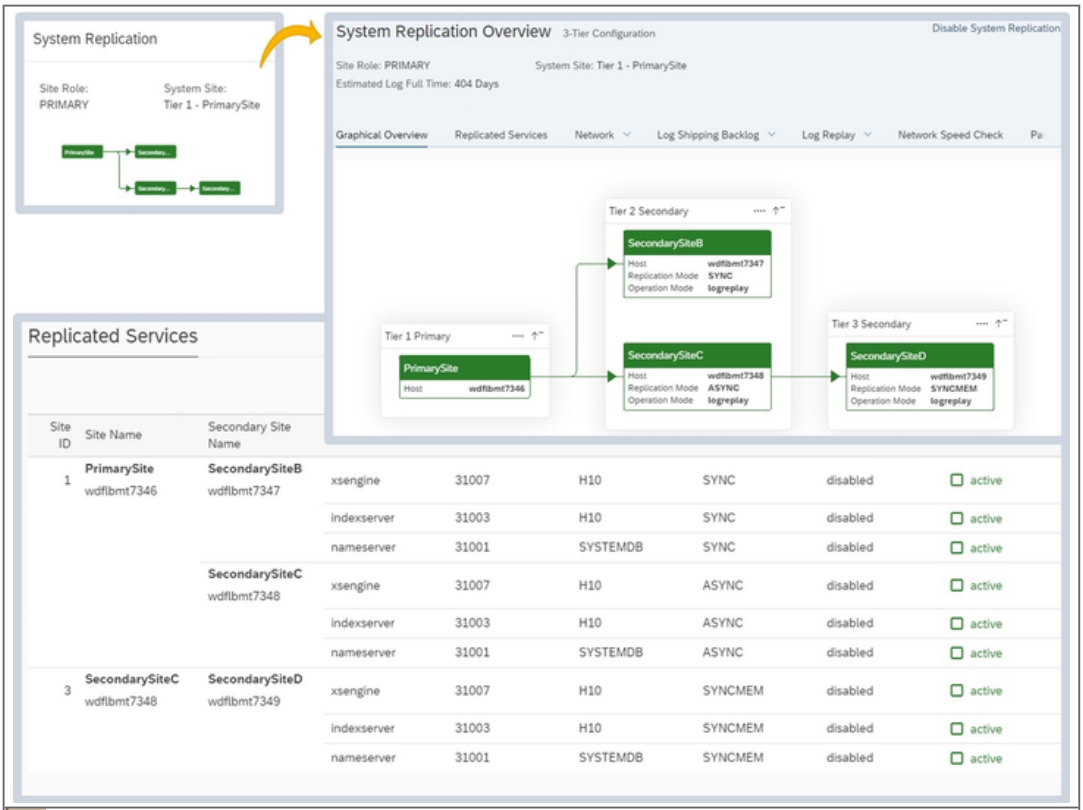


Figure 100: Multitarget Replication - Example

```
> hdbnsutil -sr_state
System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~
online: true
mode: primary
operation mode: primary
site id: 1
site name: PrimarySite

is source system: true
is secondary/consumer system: false
has secondaries/consumers attached: true
is a takeover active: false

Host Mappings:
~~~~~~~~~~~~~~
wdflbmt7194 -> [SecondarySiteD] wdflbmt7349
wdflbmt7194 -> [SecondarySiteC] wdflbmt7348
wdflbmt7194 -> [SecondarySiteB] wdflbmt7347
wdflbmt7194 -> [PrimarySite] wdflbmt7346
Site Mappings:
~~~~~~~~~~~~~~
PrimarySite (primary/primary)
    |---SecondarySiteC (async/logreplay)
    |     |---SecondarySiteD (syncmem/logreplay)
    |---SecondarySiteB (sync/logreplay)

Tier of PrimarySite: 1
Tier of SecondarySiteC: 2
Tier of SecondarySiteD: 3
Tier of SecondarySiteB: 2
...
```

Figure 101: Multitarget Replication – Output of hdbnsutil

Multitarget Replication with Read-Enabled Secondary Site

In a multitarget system replication setup, you can configure multiple secondary systems as Active/Active (read enabled). Only one of these secondary systems can be accessed using hint-based statement routing; the others must be accessed using a direct connection.

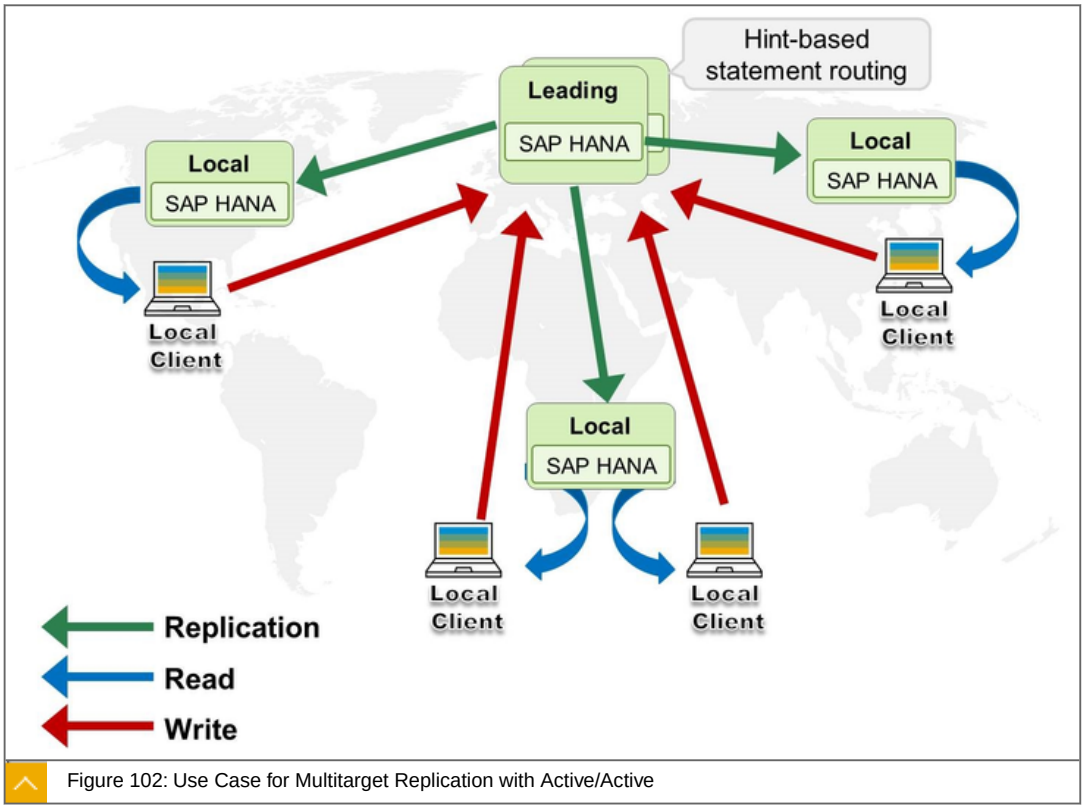


Figure 102: Use Case for Multitarget Replication with Active/Active

You can configure which of the read-enabled secondary systems is allowed for hint-based statement routing using the parameter: global.ini/[system_replication]/ hint_based_routing_site_name = <site_name> .

So far, the feature is implemented for Tier 2 systems only – Tier 3 are not enabled for this feature. Take care that systems which are supposed to use this feature are connected directly to primary system (Tier 1).

Disaster Recovery Scenarios for Multitarget System Replication

Several solutions are available when the systems involved in a multitarget system replication configuration fail.

We are using the setup described in the figure, Multitarget Replication Failure on Primary System, as an example to describe the procedure. In this setup, primary system A replicates data changes to secondary system B located in the same data center. Primary system A also replicates data changes to the secondary system C located in data center 2. Secondary system C is a source system for a further secondary system D located in the same data center with system C.

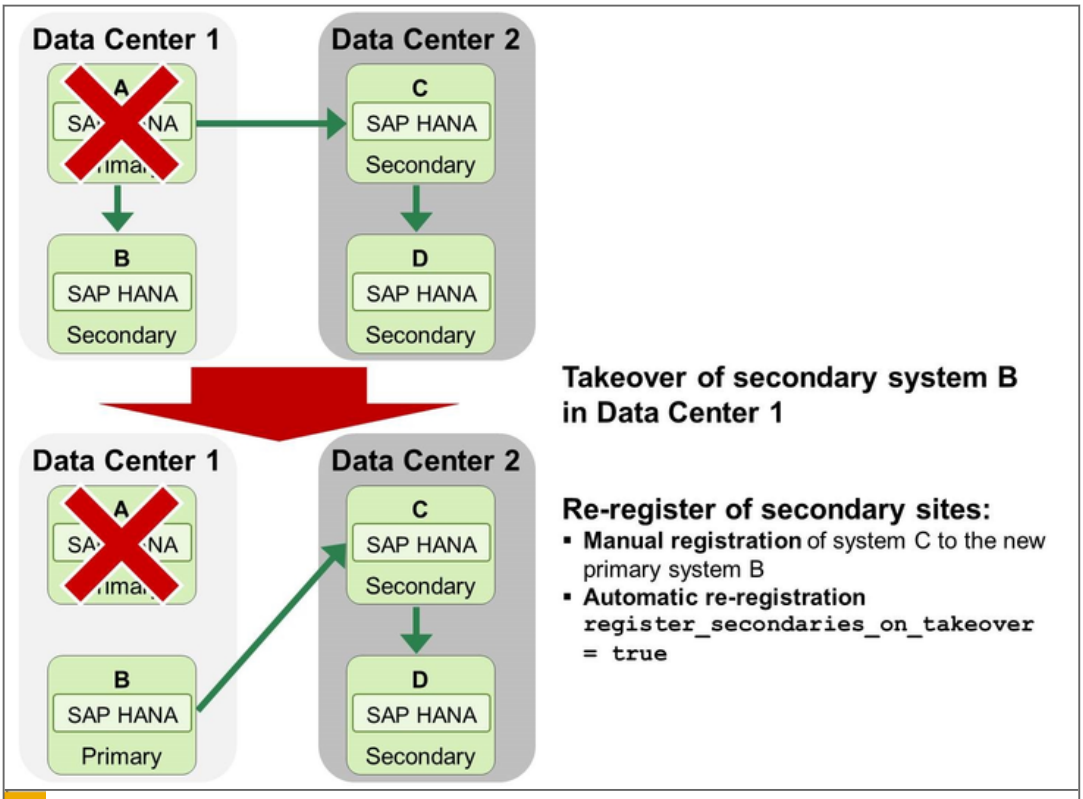


Figure 103: Multitarget Replication Failure on Primary System

When primary system A fails, proceed as follows:

1. Take over on secondary system B in data center 1.
2. Register secondary system C in data center 2 to the new primary system B in data center 1. Then, register secondary system D in data center 2 to secondary system C.
3. After the failure on the previous primary system A is solved, register it to the new primary system B in data center 1.

Alternatively, you can set the global.ini/[system_replication]/ register_secondaries_on_takeover parameter to True and take over on secondary system B in data center 1. As a result, secondary system C in data center 2 will register automatically to the new primary system B, while secondary system D in data center 2 will register automatically to secondary system C.

After the failure on the previous primary system A is solved, register it to the new primary system B in data center 1.

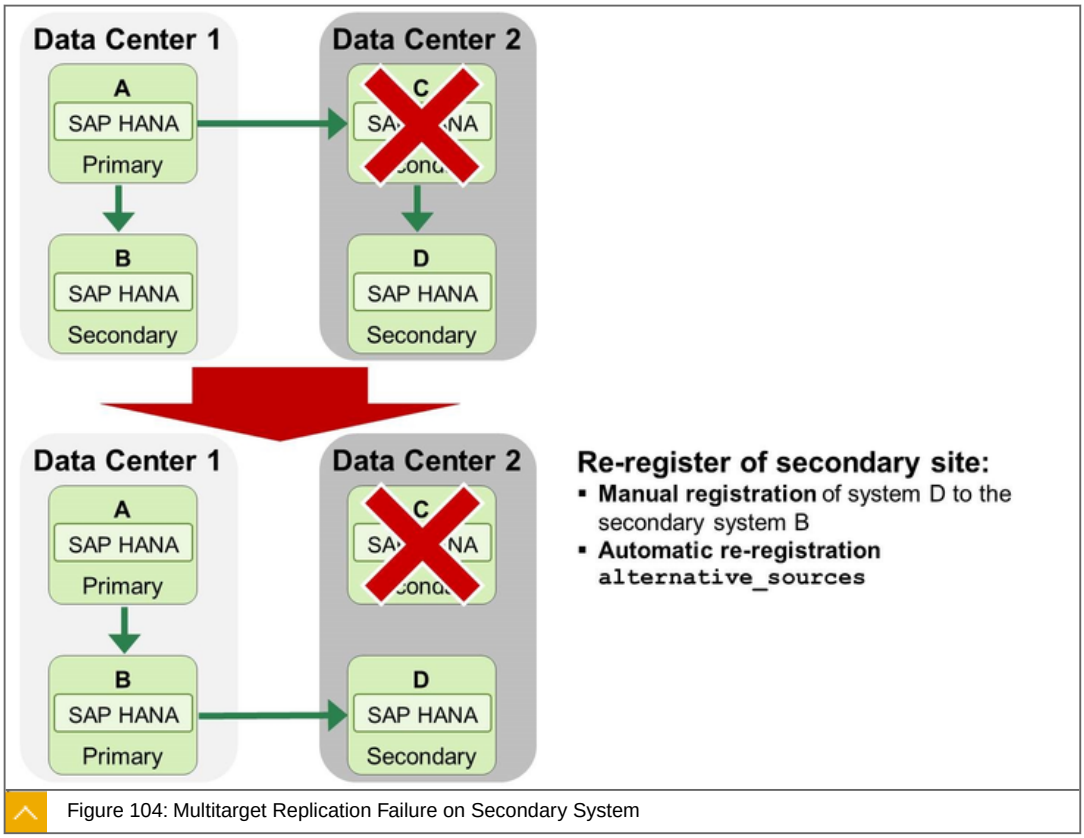


Figure 104: Multitarget Replication Failure on Secondary System

When secondary system C fails, register secondary system D to secondary system B in data center 1.

Alternatively, you can set parameters on the specific secondary so that the secondary system D will register automatically secondary system D to secondary system B in data center 1:

global.ini/[system_replication]/alternative_sources

global.ini/[system_replication]/
retries_before_register_to_alternative_source

Example: alternative_sources =SiteC:sync,SiteB:async ,..

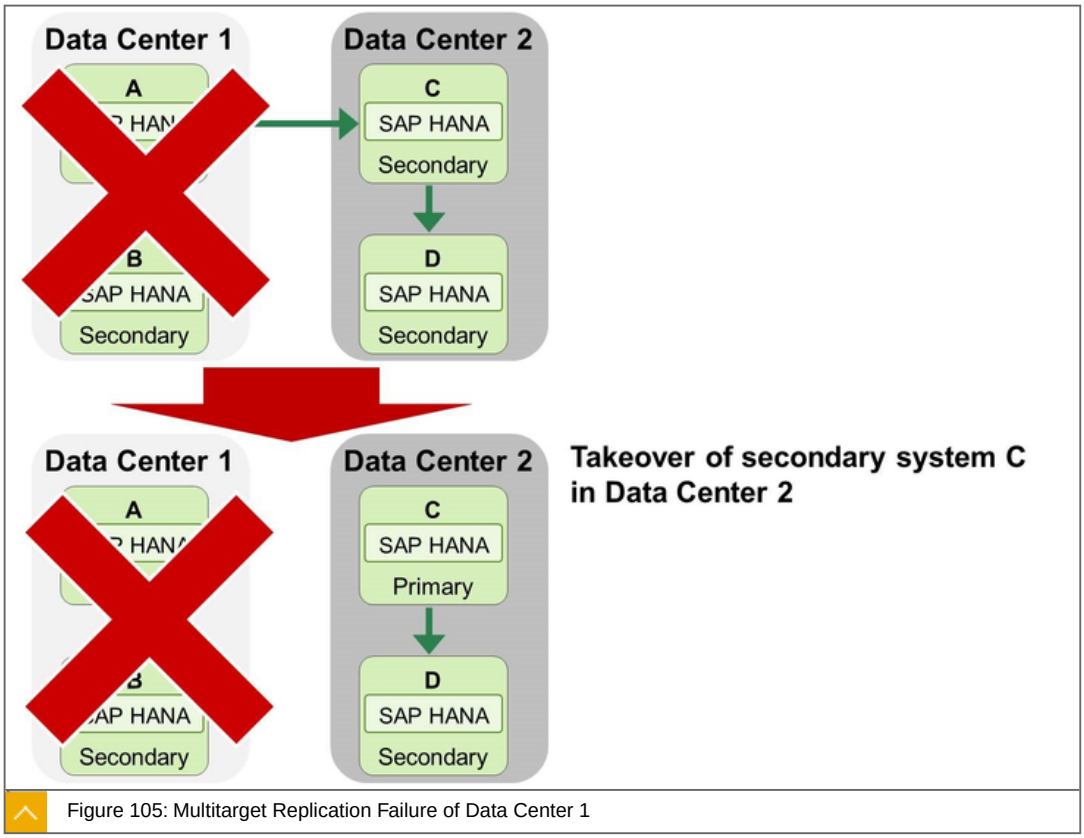


Figure 105: Multitarget Replication Failure of Data Center 1

When all the systems in data center 1 fail as shown in figure, Multi Target Replication Failure of Data Center 1, proceed as follows:

1. Take over on secondary system C in data center 2.
2. After the failure on the previous primary system is solved, register system A to the new primary system C in data center 2.
3. Register secondary system B as tier 3 to system A in data center 1.

Use Multitarget System Replication for Near Zero Downtime Upgrades

You can upgrade your SAP HANA systems running in a multitarget system replication setup.

Example:

Primary system A replicates data changes to secondary system B located in the same data center. Primary system A also replicates data changes to the secondary system C located in data center 2. Secondary system C is a source system for a further secondary system D located in the same data center.

In this setup:

The primary system is the production system.

The secondary system located in the same data center as the primary system will become the production system after the upgrade. Further secondary systems are located in a remote data center.

There is no replication error.

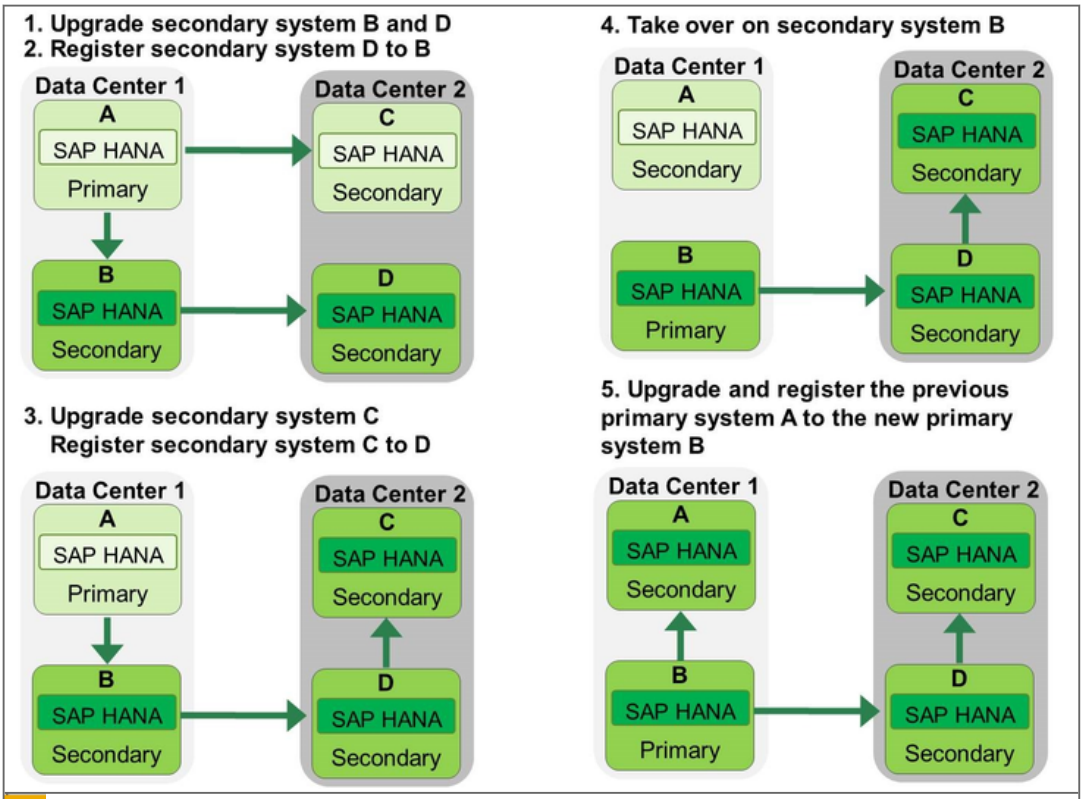


Figure 106: MultiTarget Replication - Upgrade

LESSON SUMMARY
You should now be able to:

Explain Multitier and Multitarget System Replication

Learning Assessment

1. With SAP HANA storage replication, you can use the servers on the secondary system for non-production SAP HANA systems.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

2. When using the Continuous Log Replay operation mode (logreplay), the shipped redo log is continuously replayed on the secondary site and read-only access on the secondary site is allowed.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

3. Which of the following are prerequisites for System Replication with QA and Development System on the secondary site?

   Choose the correct answers.

   - ☐ **A** Preload of tables must be switched off on the secondary.
   - ☐ **B** The instance numbers for the replicated system and the Development/QA system are identical.
   - ☐ **C** Development/QA systems need to be shut down in case of a takeover.
   - ☐ **D** Additional independent disk volume is needed for Development/QA systems.

4. Which tool can you use to set up SAP HANA system replication in one step?

   Choose the correct answer.

   - ☐ **A** SAP HANA Studio
   - ☐ **B** SAP HANA Web IDE
   - ☐ **C** SAP HANA Cockpit
   - ☐ **D** hdbnsutil

5. The system replication status **Initializing** indicates that the initial data transfer is in progress.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

6. You create a new tenant database in a configured SAP HANA system replication. Which prerequisite must be fulfilled for the new tenant database to participate in the replication?

   Choose the correct answer.

   - ☐ **A** It must be stopped.
   - ☐ **B** It must be backed up.
   - ☐ **C** A fallback snapshot must be created
   - ☐ **D** The restart mode must be set to No auto-restart

7. Which tool can you use to trigger a takeover from the current primary to the secondary system?

   Choose the correct answers.

   - ☐ **A** hdblcm
   - ☐ **B** hdbsql
   - ☐ **C** SAP HANA Cockpit
   - ☐ **D** hdbnsutil

8. System replication with a read-enabled secondary site is based on the delta data shipping operation mode.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

9. Which replication modes are supported for system replication with a read-enabled secondary site?

   Choose the correct answers.

   - ☐ **A** SYNC
   - ☐ **B** Full SYNC
   - ☐ **C** ASYNC
   - ☐ **D** SYNCMEM

10. After an invisible takeover, the client keeps the connections to the primary system and the sessions are restored to the secondary system.

    Determine whether this statement is true or false.

    - ☐ True
    - ☐ False

11. Enabling SAP HANA system replication with secondary time travel has no impact on the sizing and performance of the secondary system.

    Determine whether this statement is true or false.

    - ☐ True
    - ☐ False

12. You can use system replication to upgrade your SAP HANA systems because the secondary system can run with a higher software version than the primary system.

    Determine whether this statement is true or false.

    - ☐ True
    - ☐ False

13. To which use cases does multitarget system replication bring advantages?

    Choose the correct answers.

    - ☐ **A** Update scenarios
    - ☐ **B** Reaching higher availability
    - ☐ **C** Necessity to perform backups
    - ☐ **D** Rearrangements of system replication multitier chains

Learning Assessment - Answers

1. With SAP HANA storage replication, you can use the servers on the secondary system for non-production SAP HANA systems.

   Determine whether this statement is true or false.

   [X] True

   [ ] False

   You are correct! Depending on the hardware solution, you can use the servers of the secondary system for other SAP HANA systems (such as test systems) until the takeover.

2. When using the Continuous Log Replay operation mode (logreplay), the shipped redo log is continuously replayed on the secondary site and read-only access on the secondary site is allowed.

   Determine whether this statement is true or false.

   [ ] True

   [X] False

   You are correct! When using the Continuous Log Replay operation mode (logreplay), read-only access on the secondary site is not allowed. Therefore, you have to configure the Continuous Log Replay with Active/Active operation mode (logreplay_readaccess).

3. Which of the following are prerequisites for System Replication with QA and Development System on the secondary site?

   Choose the correct answers.

   [X] **A** Preload of tables must be switched off on the secondary.

   [ ] **B** The instance numbers for the replicated system and the Development/QA system are identical.

   [ ] **C** Development/QA systems need to be shut down in case of a takeover.

   [X] **D** Additional independent disk volume is needed for Development/QA systems.

   You are correct! To use the secondary site in a system replication scenario for running Development/QA systems, the preload of tables must be switched off on the secondary site and additional independent disk volume is needed for Development/QA systems.

4. Which tool can you use to set up SAP HANA system replication in one step?

   Choose the correct answer.

   - [ ] **A** SAP HANA Studio
   - [ ] **B** SAP HANA Web IDE
   - [X] **C** SAP HANA Cockpit
   - [ ] **D** hdbnsutil

   You are correct! SAP HANA Cockpit allows you to enable the primary system and then register the secondary system from the primary system in one step.

5. The system replication status **Initializing** indicates that the initial data transfer is in progress.

   Determine whether this statement is true or false.

   - [X] True
   - [ ] False

   You are correct! The system replication status Initializing indicates that the initial data transfer is in progress. In this state the secondary system is not usable.

6. You create a new tenant database in a configured SAP HANA system replication. Which prerequisite must be fulfilled for the new tenant database to participate in the replication?

   Choose the correct answer.

   - [ ] **A** It must be stopped.
   - [X] **B** It must be backed up.
   - [ ] **C** A fallback snapshot must be created
   - [ ] **D** The restart mode must be set to No auto-restart

   You are correct! If a new tenant database is created in a configured SAP HANA system replication, it must be backed up to participate in the replication.

7. Which tool can you use to trigger a takeover from the current primary to the secondary system?

   Choose the correct answers.

   - [ ] **A** hdblcm
   - [ ] **B** hdbsql
   - [X] **C** SAP HANA Cockpit
   - [X] **D** hdbnsutil

   You are correct! The takeover from the current primary to the secondary system can be triggered using SAP HANA Cockpit and hdbnsutil.

8. System replication with a read-enabled secondary site is based on the delta data shipping operation mode.

   Determine whether this statement is true or false.

   - [ ] True
   - [X] False

   You are correct! System replication with a read-enabled secondary site is based on the continuous log replay operation mode.

9. Which replication modes are supported for system replication with a read-enabled secondary site?

   Choose the correct answers.

   - [X] **A** SYNC
   - [X] **B** Full SYNC
   - [X] **C** ASYNC
   - [X] **D** SYNCMEM

   You are correct! All existing replication modes are supported for system replication with a read-enabled secondary site.

10. After an invisible takeover, the client keeps the connections to the primary system and the sessions are restored to the secondary system.

    Determine whether this statement is true or false.

    - [X] True
    - [ ] False

    You are correct! After an invisible takeover, the client keeps the connections to the primary system and the sessions are restored to the secondary system.

11. Enabling SAP HANA system replication with secondary time travel has no impact on the sizing and performance of the secondary system.

    Determine whether this statement is true or false.

    ☐ True

    ☒ False

    You are correct! For time travel to work, log information and snapshots are kept online in the data area. Because of this, log information and data grows on the secondary system when time travel is turned on.

12. You can use system replication to upgrade your SAP HANA systems because the secondary system can run with a higher software version than the primary system.

    Determine whether this statement is true or false.

    ☒ True

    ☐ False

    You are correct! SAP HANA offers zero downtime maintenance, together with SAP HANA system replication. You can use system replication to upgrade your SAP HANA systems because the secondary system can run with a higher software version than the primary system.

13. To which use cases does multitarget system replication bring advantages?

    Choose the correct answers.

    ☒ **A** Update scenarios

    ☒ **B** Reaching higher availability

    ☐ **C** Necessity to perform backups

    ☒ **D** Rearrangements of system replication multitier chains

    You are correct! Multitarget system replication can bring advantages for update scenarios, for reaching higher availability, and for rearrangements of system replication multitier chains.

UNIT 4

SAP HANA Tenant Replication

Lesson 1

UNIT OBJECTIVES

Understand the setup of tenant replication

Unit 4
Lesson 1

Explaining Tenant Replication

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand the setup of tenant replication

Copying and Moving Tenant Databases Between Systems

Using SAP HANA system replication mechanisms, SAP HANA tenant databases 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.

Copying and moving a tenant database are essentially the same processes. 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.

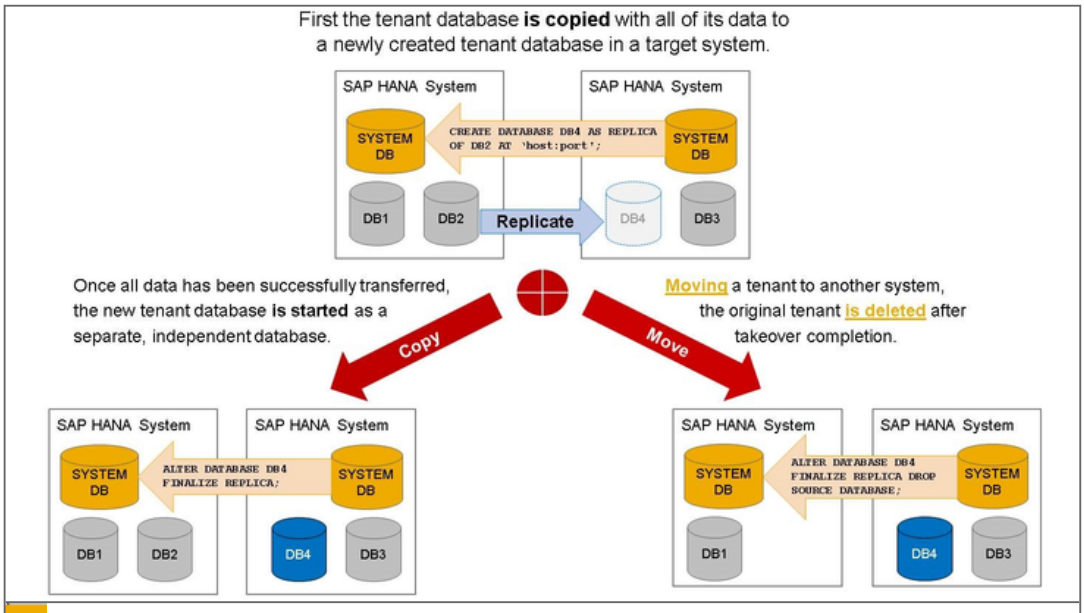


Figure 107: Copying and Moving Tenant Databases

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 than the source system.

You can also use this mechanism to create a copy of a tenant database on the same host or to copy/move a tenant database to another host that is part of the same system.

Use Cases for Copying and Moving Tenants

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.

Prerequisites and Implementation Considerations

A new tenant database will be created 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.

During the copy and move process, data must be replicated using a secure (SSL/TLS) network connection by default.

In a running system replication, it is possible to copy or move tenant databases into a primary system, or from a primary system into another target system that is different than the secondary system.

There can be no changes to the topology of the original tenant database while the move or copy is in progress.

If the source system is configured for host auto-failover, the copy or process fails in the event of failover to a standby host.

The administrator has the system privilege DATABASE ADMIN.

Tenant replication is configured.

A complete system backup for the source tenant database has been created.

Create a Tenant Database Using the SAP HANA Replication Mechanism

The figure Create Tenant Using Replication describes how to create a new tenant, based on an existing tenant using the SAP HANA replication mechanism on a target system performed with the SAP HANA Cockpit tool.

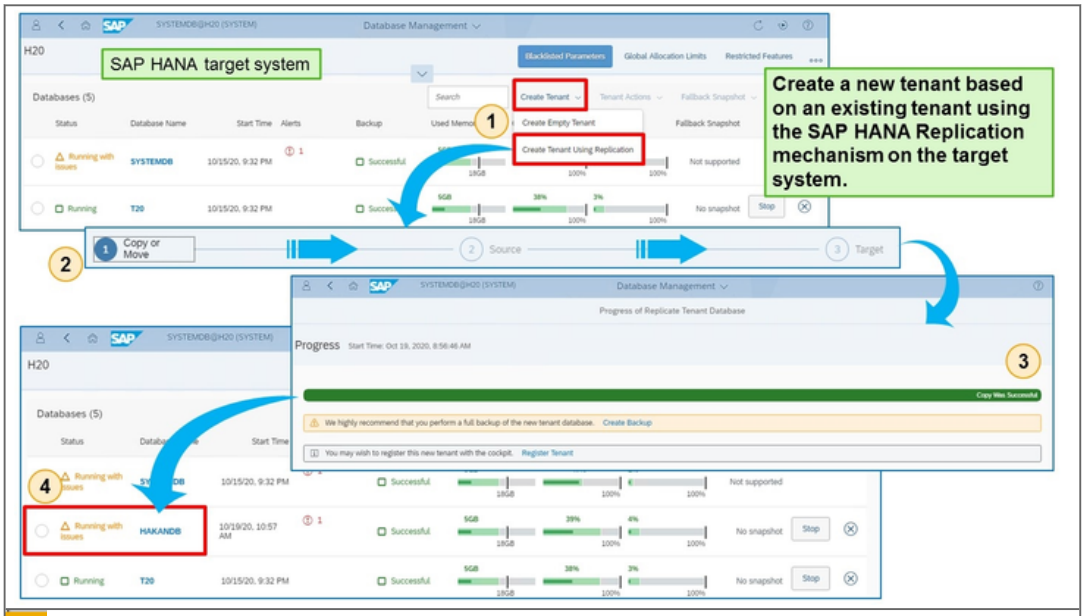


Figure 108: Create Tenant Using Replication

1. To create a new tenant based on an existing tenant, on the Database Management screen of your SAP HANA target systems SYSTEMDB, choose Create Tenant Create Tenant Using Replication.
2. Follow the guided steps. To create a copy of an existing tenant database, select Copy using replication . To remove the original tenant after the copy has been created, select Move using replication . Choose the source system- and tenant databases and enter a name for the new tenant.
3. Perform the following steps:

   Optional: Specify the number of the internal communication port of the listed services.

   (For high isolation systems only) Enter a dedicated OS user and group for the source.

   If configuration changes are required for the replication, warnings appear to indicate the required changes. Select Approve to proceed.

   If prompted, enter the SAP Control credentials required for the restart of the system.

   If a trust relationship has not yet been established between the source system and the target system, select an existing public key certificate or upload a certificate.

   Review the summary, then choose Copy Tenant Database or Move Tenant Database , to start replicating the tenant database.
4. The new tenant database appears in the list of databases of your target systems SYSTEMDB.

Note:
For detailed information about creating tenant databases using replication, please check the SAP HANA Administration Guide and the SAP HANA Administration with SAP HANA Cockpit document.

Copy a Tenant Database Using the SAP HANA Replication Mechanism

The figures Copy Tenant Using Replication (1, 2, and 3) describe how to copy a tenant database to another system using the SAP HANA replication mechanism and the SAP HANA Cockpit tool.

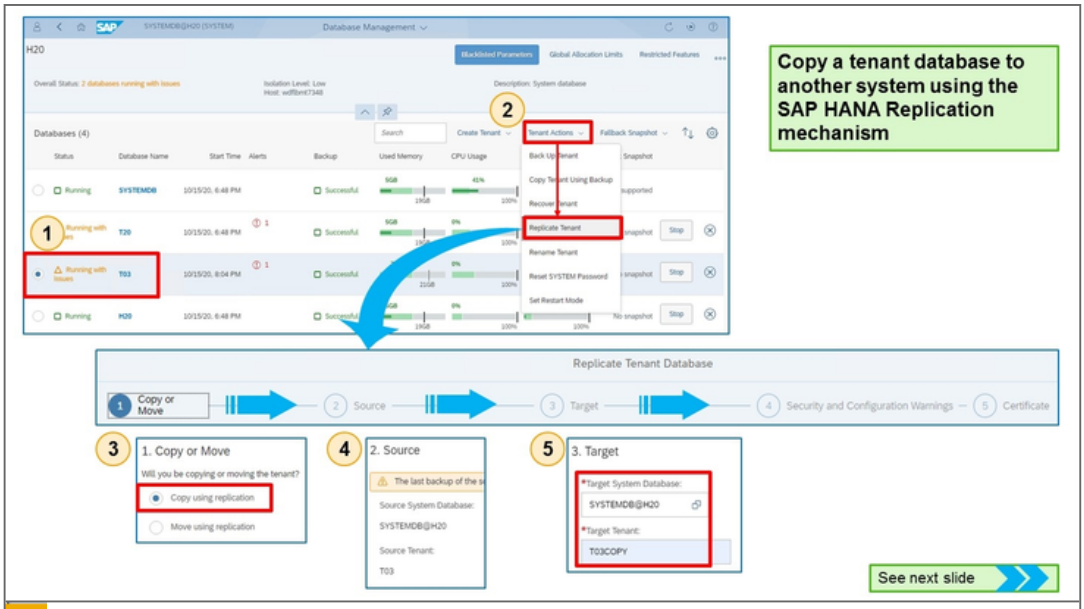


Figure 109: Copy Tenant Using Replication (1/3)

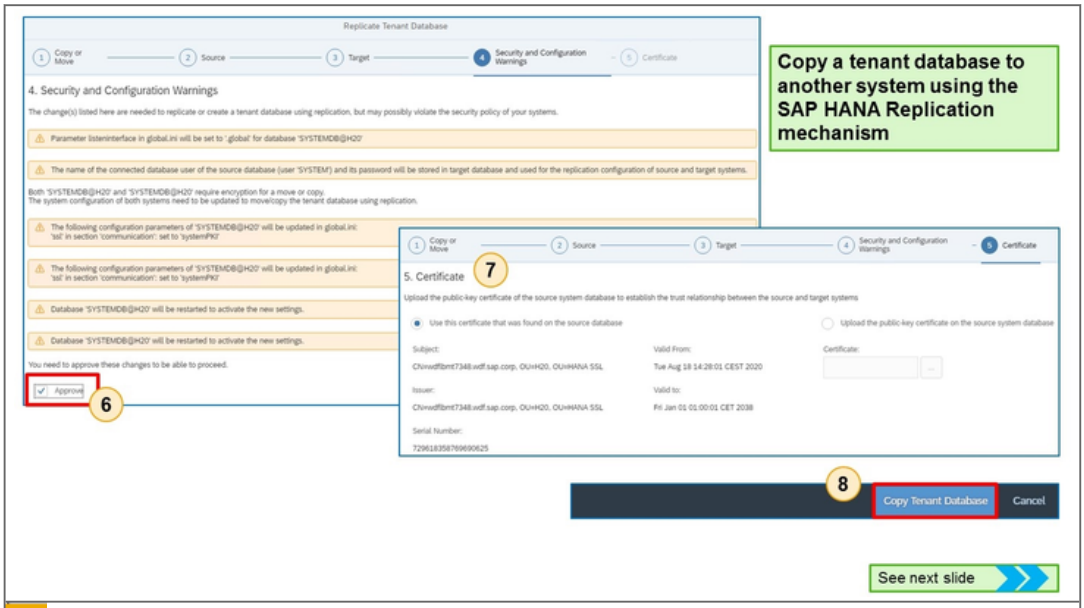


Figure 110: Copy Tenant Using Replication (2/3)

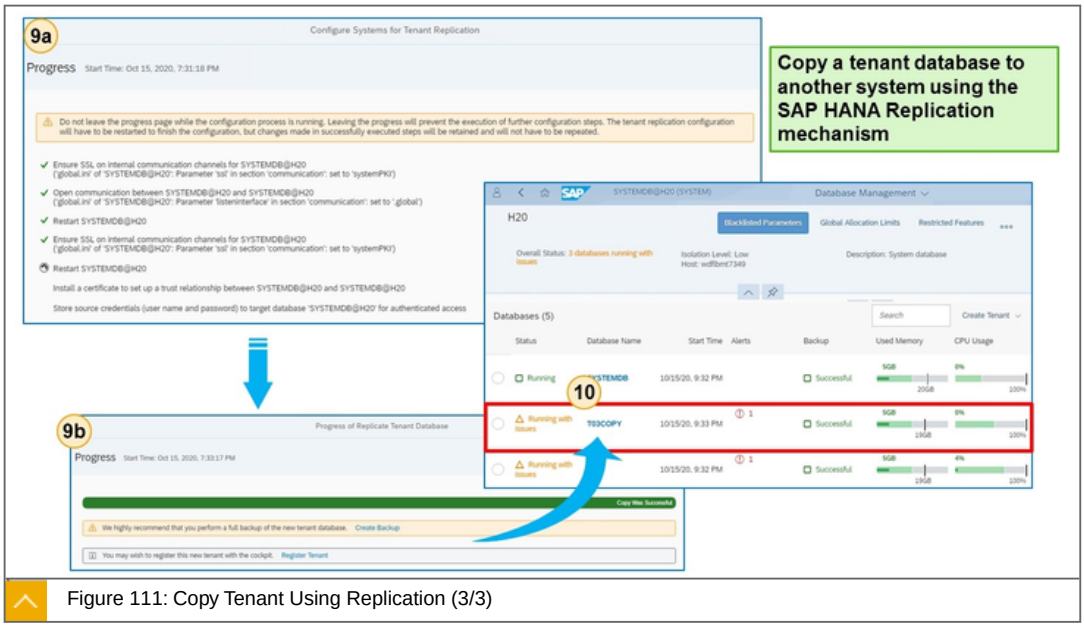


Figure 111: Copy Tenant Using Replication (3/3)

1. On the Database Management screen of your SAP HANA source systems SYSTEMDB, mark your source tenant.
2. Choose Tenant Actions Replicate Tenant Using Replication .
3. Choose Copy using replication .
4. The Source System Database and the Source Tenant will be provided automatically.
5. Choose the Target System Database and enter a name for the Target Tenant .

   Optional: Under Advanced Settings, specify the port number for each service. If you do not enter a port, it is assigned automatically based on port number availability. In multihost systems enter the host and port of a service.

   (For high isolation systems only) Enter a dedicated OS user and group for the source.
6. If configuration changes are required for the replication, warnings appear to indicate the required changes. Select Approve to proceed.

   If prompted, enter the SAP Control credentials required for the restart of the system.
7. If a trust relationship has not yet been established between the source system and the target system, select an existing public key certificate or upload a certificate.
8. Review the summary, then choose Copy Tenant Database , to start replicating the tenant database.
9. Observe the following:
   a. The Progress page shows the running configuration process, the system restarts, and the certification installation.
   b. After finishing, the progress bar indicates the status of the tenant replication.
10. The copied tenant database appears in the Database Management screen of the target systems SYSTEMDB.

Move a Tenant Database Using the SAP HANA Replication Mechanism

The figures Move Tenant Using Replication (1 and 2) describe how to move a tenant database to another system using the SAP HANA replication mechanism and the SAP HANA Cockpit tool.

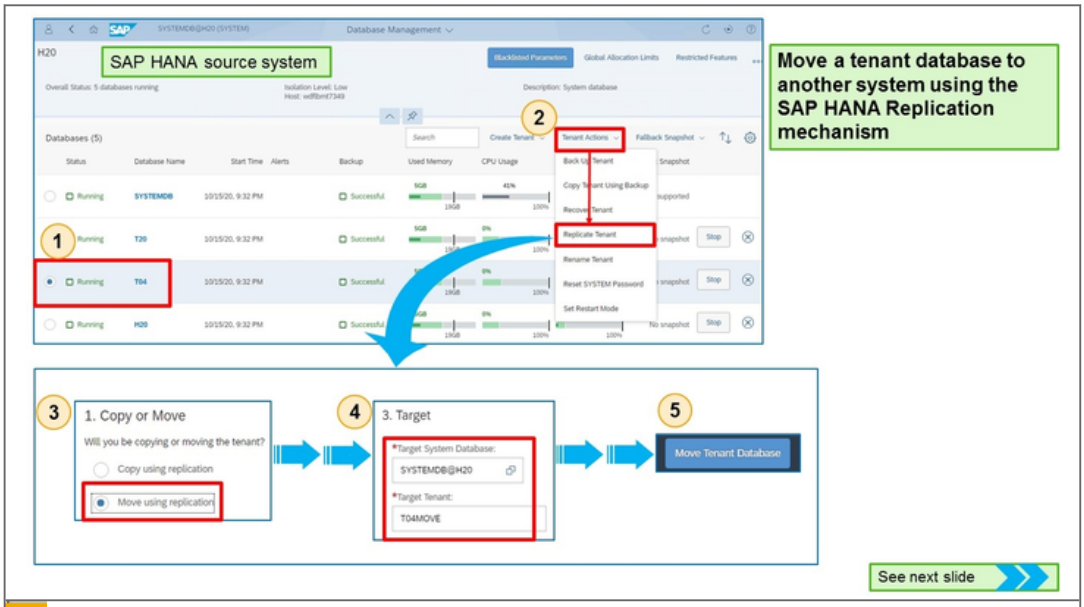


Figure 112: Move Tenant Using Replication (1/2)

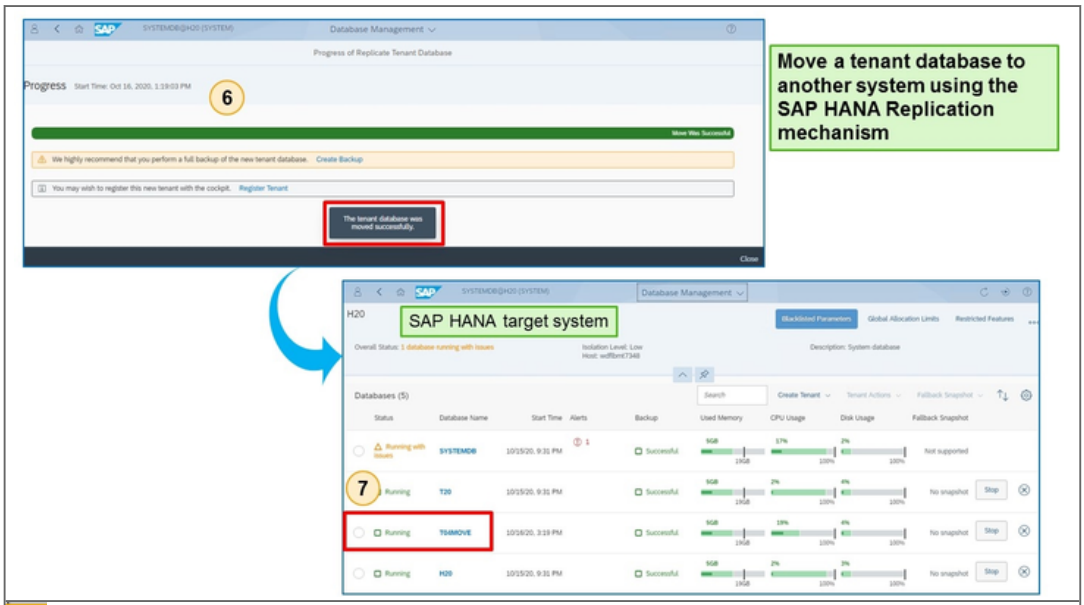


Figure 113: Move Tenant Using Replication (2/2)

As mentioned before, copying and moving a tenant database are essentially the same process.

The differences are:

In step 4), choose Move using replication . This will remove the original tenant after the copy has been created.

In step 5), the button is named Move Tenant Database .

Note:
The process of copying and moving tenant databases is described in detail in the SAP HANA Administration Guide and the SAP HANA Administration with SAP HANA Cockpit document.

Copy and Move Process

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

1. To prepare the copy and move process, the system administrator performs the following tasks:

   Verifies that TLS/SSL is enabled on internal communication channels.

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

   Creates credentials for authenticated access to the source system.

   Configures a secure connection from the target system to source system.

   Backs up the source tenant database.

2. The 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.

3. During the copy process, the system database of the target system performs the following tasks:

   Establishes a secure connection to the system database using the stored credentials created earlier.

   Creates a new tenant database with the same topology as the tenant database in the source system.

   Initiates replication of data between the services in the source tenant database and the corresponding services in the target database.

   Commits the copy by executing the SQL statement ALTER DATABASE FINALIZE REPLICA when the replication status is ACTIVE.

4. To finalize the copy and move process, the following steps are performed:

   The system database of the target system starts the target tenant database and performs clean-up operations.

   The system administrator performs manual post-copy or post-move steps.

Setup for the Exercise

In the following exercise, you will copy and move a tenant database from a source system to a target system using the SAP HANA system replication mechanism.

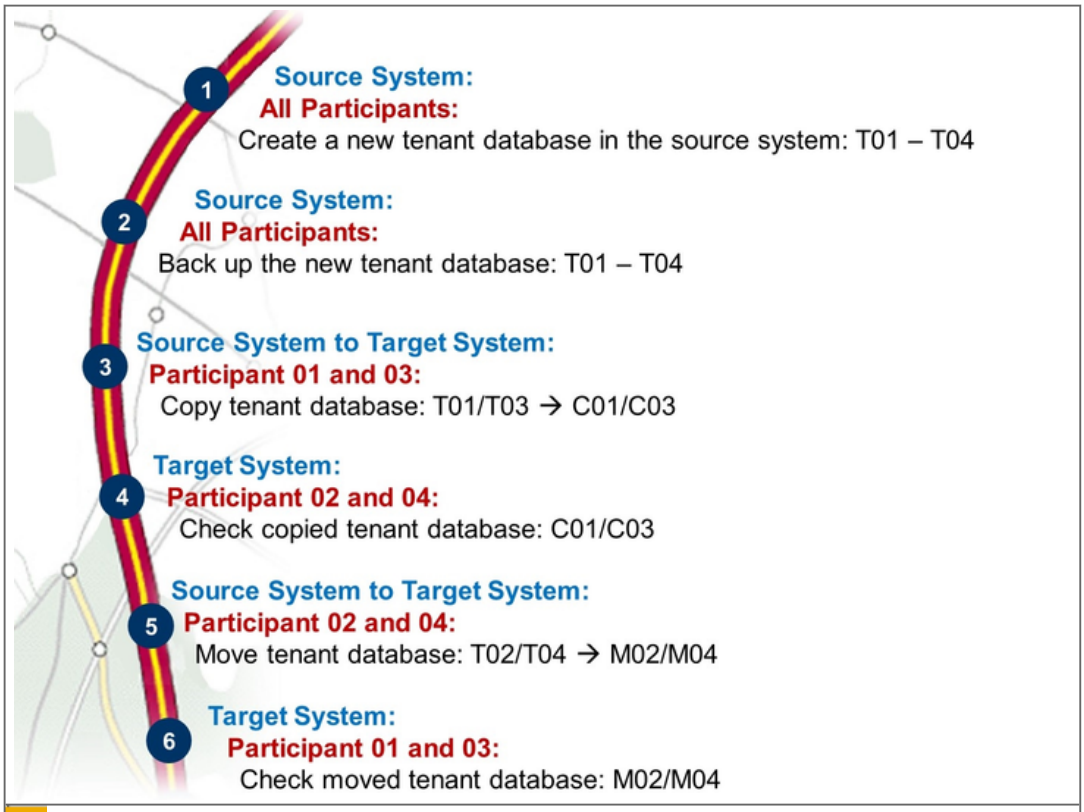


Figure 114: Exercise: Copy and Move a Tenant Database Between Systems

LESSON SUMMARY
You should now be able to:

- Understand the setup of tenant replication

Learning Assessment

1. Moving tenant databases can be used for a tenant database-specific upgrade.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

Learning Assessment - Answers

1. Moving tenant databases can be used for a tenant database-specific upgrade.

   Determine whether this statement is true or false.

   [X] True

   [ ] False

   You are correct! If you want to upgrade a single tenant database but not the entire system, you move the tenant database to a system already running the higher version.

UNIT 5

Appendix: HANA Additional Scripts

UNIT OBJECTIVES

Understand the Python support scripts used in SAP HANA

Reinitialize a non-recoverable system database

Unit 5
Lesson 1

Appendix: Using Python Support Scripts in SAP HANA

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Understand the Python support scripts used in SAP HANA

Python Support Scripts in SAP HANA

Business Example

As an SAP HANA database administrator, sometimes you need to access the SAP HANA database directly from the command line, without the use of the SAP HANA Cockpit. There are several useful Python scripts that you can use at the command line to access configuration and monitor data stored in the SAP HANA database.

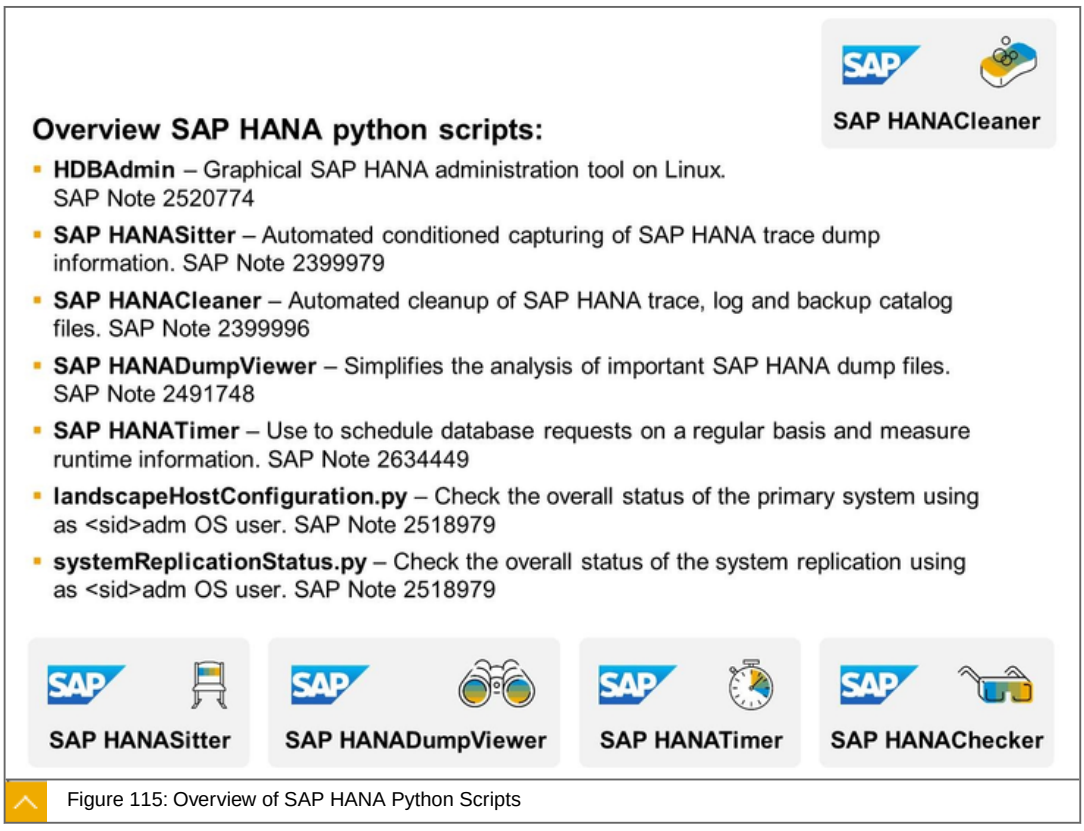


Figure 115: Overview of SAP HANA Python Scripts

The Support Tool HDBAdmin.sh

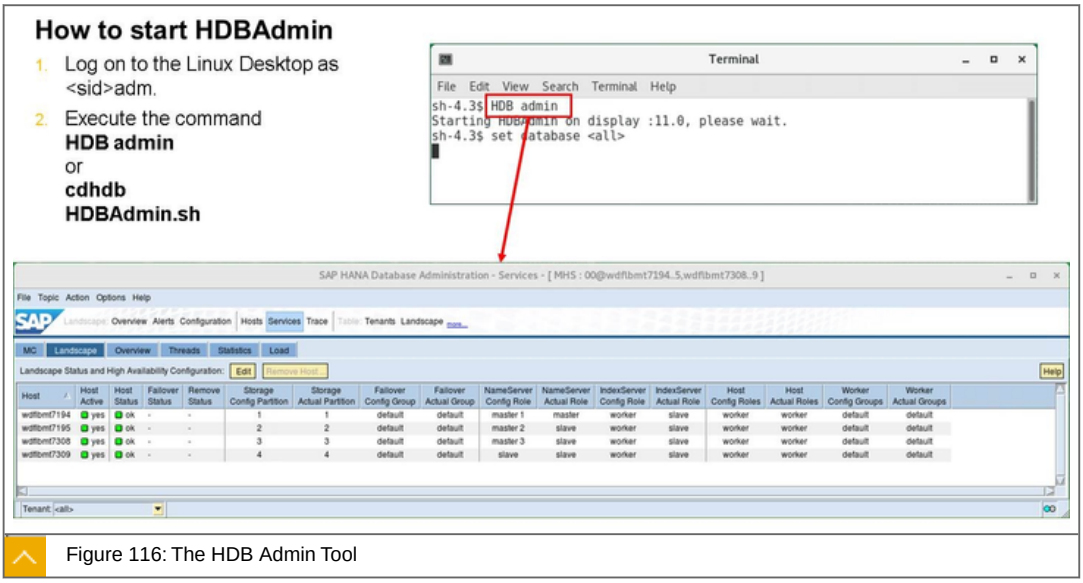


Figure 116: The HDB Admin Tool

During the exercises, we used the HDBAdmin tool, therefore more explanation about this tool may be useful.

This tool was created for internal use by SAP and is not supported for public use. You can use the tool if you wish, but SAP does not provide support for issues encountered while using the tool or the result set produced by the tool.

During our exercises, the HDBAdmin tool was very useful because it had the one-second automatic refresh feature, so we could see the host auto-failover happen in real time.

For daily business, SAP recommends using the SAP HANA Cockpit 2.0 for system administration and performance monitoring tasks.

The HDBAdmin tool has no documentation, which sometimes makes it difficult to use. Its origin is the standalone TREX Admin tool for the SAP BW Accelerator. In that area, there is some documentation. Also, the HDBAdmin tool is used and shown in several demo videos in the SAP HANA Academy.

Starting the HDBAdmin Tool

Log on to the Linux desktop as the <sid>adm user to start the HDBAdmin tool. On Linux, the HDB Admin tool has a graphical interface, therefore you need an X server installed on your Linux server. This is often seen as a problem by Unix administrators, so starting HDBAdmin locally is difficult and you cannot use a terminal program that only supports text mode, such as ssh.

A solution to this problem is simple. Because the X Window System is an architecture-independent system for remote graphical user interfaces over the network, you can use your Windows PC or laptop as a graphical display.

Microsoft Windows by default has no X Windows support, but there are many third-party implementations. Several of these implementations are free and open source. Some examples are Cygwin/X and Xming. Others are paid and proprietary, such as Exceed, MKS X/Server, and Reflection X.

> Caution:
> Always keep in mind that you use the HDBAdmin tool at your own risk, and SAP does not provide support for issues encountered while using the tool.

Useful SAP Notes

SAP Note: 2534881 - Issues while working with HDBAdmin tool

The hanasitter.py Script

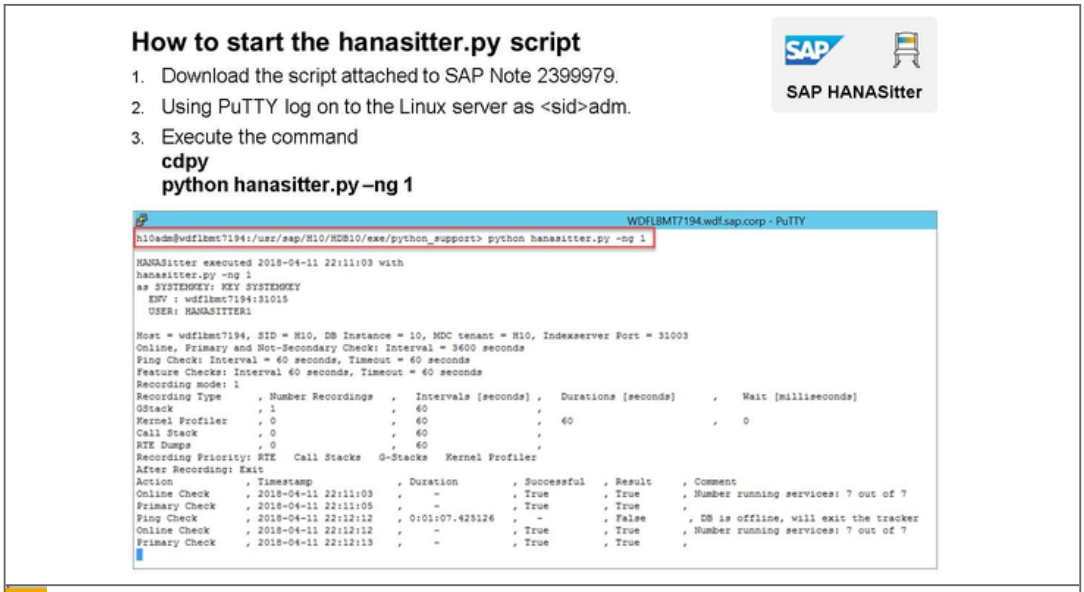


Figure 117: The hanasitter.py Script

HANASitter Features:

- Database online check
- CPU, ping, and critical feature check
- Recording mode for RTE dumps, stack calls, kernel profiler trace, and GStack
- Scale-out monitor
- Critical session killer

The SAP HANA sitter checks by default once an hour, if SAP HANA is online and primary. In this case, it starts to track. Tracking includes regularly (by default one minute) checking if SAP HANA is responsive. If it is not, it starts to record.

Recording can include writing call stacks of all active threads, recording run time dumps, index server gstacks, and/or kernel profiler traces. By default, nothing is recorded.

If SAP HANA is responsive, it checks many of the critical features of SAP HANA. As standard, the script checks if there are more than 30 active threads. If there are more than 30 active threads, the script starts to record.

When the script has finished recording, it exits. The script can be configured to restart from the command line.

When the script has finished all the tests successfully, it sleeps for one hour, before it starts all the checks again.

Setup Steps Overview

1. Create an SAP HANA user (for example, HANASITTER, but you can use a different name) and assign the CATALOG READ privilege.
2. Create a user key (for example, SYSTEMKEY, but you can use a different name) in the hdbuserstore.
3. Download the hanasitter.py script attached to SAP Note: 2399979.
4. Store the script in, for example, the python_support directory.
5. As <sid>adm, change to the python_support directory with the command **cdpy**.
6. Execute the script with the command **python hanasitter.py -ng 1** .

Useful SAP Notes

SAP Note: 2399979 - How-To: Configuring automatic SAP HANA Data Collection with SAP HANASitter

The hanacleaner.py Script

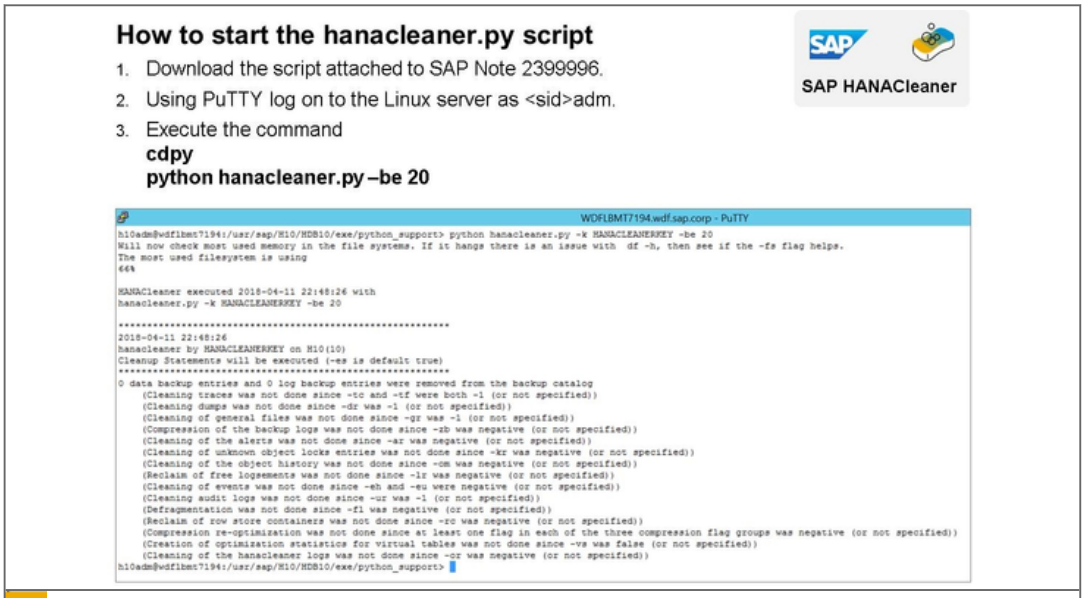


Figure 118: The hanacleaner.py Script

The SAP HANA cleaner is a housekeeping service for SAP HANA. It can be used to clean the backup catalog, diagnostic files, and alerts, and to compress the backup logs. It should be executed by <sid>adm or, if you use a CRON job, with the same environment as <sid>adm. For more information, see SAP Note: 2399996 and SAP Note: 2400024.

HANACleaner Features:

- Cleanup for backup catalog and backup logs
- Cleanup for trace and dump files
- Log segment cleanup

Audit log cleanup

Cleanup old alerts from the alert table

Remove old ini file history

Re-optimize table compression

Check and repair disk fragmentation

Setup Steps Overview

1. Create an SAP HANA user (for example, HANACLEANER, but you can use a different name) and assign the CATALOG READ privilege.
2. Create a user key (for example, SYSTEMKEY, but you can use a different name) in the hdbuserstore.
3. Download the hanacleaner.py script attached to SAP Note: 2399996.
4. Store the script in, for example, the python_support directory.
5. As <sid>adm, change to the python_support directory with the command **cdpy**.
6. Execute the script with the command **python hanacleaner.py -be 20** .

Useful SAP Notes

SAP Note: 2399996 - How-To: Configuring automatic SAP HANA Cleanup with SAP HANACleaner

SAP Note: 2400024 - How-To: SAP HANA Administration and Monitoring

Download Locations for Additional SAP HANA Scripts

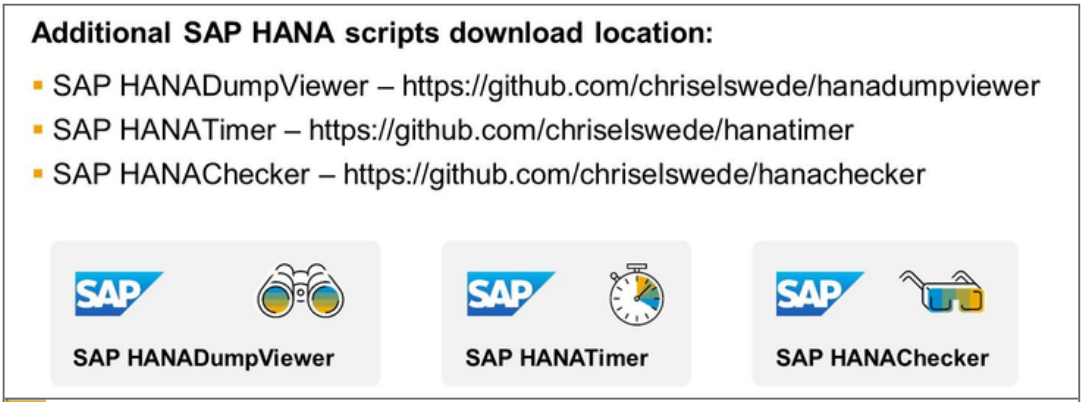


Figure 119: Additional Python Scripts and Locations

SAP HANACleaner, SAP HANASitter, SAP HANADumpViewer, SAP HANATimer, and SAP HANAChecker are very useful tools for automating several SAP HANA administration tasks. These tools are provided “as is” and can be found at Github. The generic Github link is: https://github.com/chriselswede .

Before using the tools, please read this disclaimer:

The SAP HANA Tools are **not** SAP official software, so normal SAP support of SAP HANA Tools cannot be assumed.

The SAP HANA Tools are open source.

The SAP HANA Tools are provided "as is".

The SAP HANA Tools are to be used at "your own risk".

The SAP HANA Tools are a one-man's hobby; developed, maintained and supported only during non-working hours.

Please read all the SAP HANA Tools documentation before using the tools.

The landscapeHostConfiguration.py Script

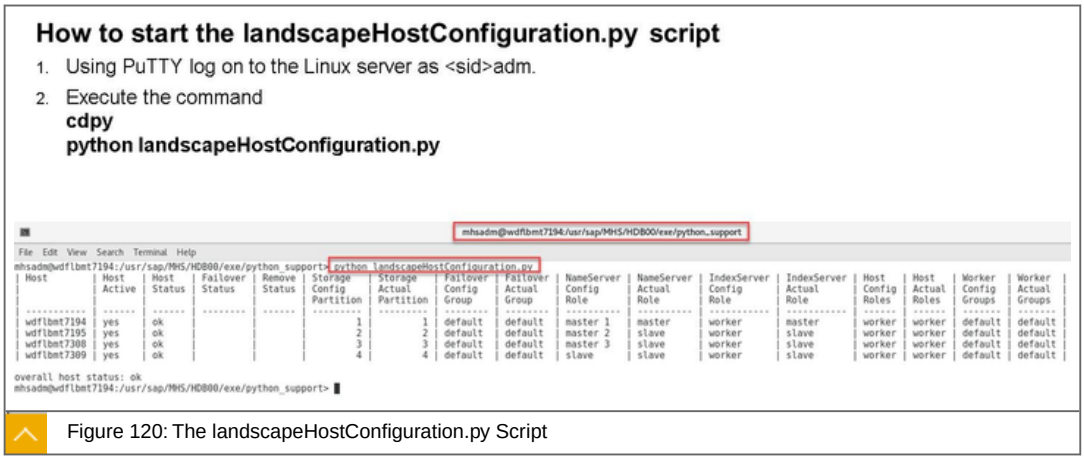


Figure 120: The landscapeHostConfiguration.py Script

During the exercises in Unit 2 and Unit 4, we used the script landscapeHostConfiguration.py several times. This script is very useful to quickly get a complete overview of the landscape setup of a multi-host SAP HANA system.

As the OS user <sid>adm, you can use the script landscapeHostConfiguration.py in the python_support folder to display topology information. Unlike the monitoring views, this approach can also be used when SAP HANA is down.

As found in the help documentation of the landscapeHostConfiguration.py script, it can be run in three different modes.

```
landscapeHostConfiguration.py [--localhost] [--sapcontrol=1]
```

1. Running the script without an additional command line option gives the overview as shown in the previous figure.
2. Running the script with the optional command line option --localhost shows only the configuration of the host where the script was started.
3. Running the script with the optional command line option --sapcontrol=1 shows a detailed output of all landscape configuration settings and environment variables for each host. This is very useful when troubleshooting a failover problem using the command line.

Useful SAP Notes

SAP Note: 2000003 - FAQ: SAP HANA

SAP Note: 2340501 - Prohibit execution of hdbnsutil, systemReplicationStatus.py and landscapeHostConfiguration.py as root user

SAP Note: 2518979 - HANA : how to check system replication status

The systemReplicationStatus.py Script

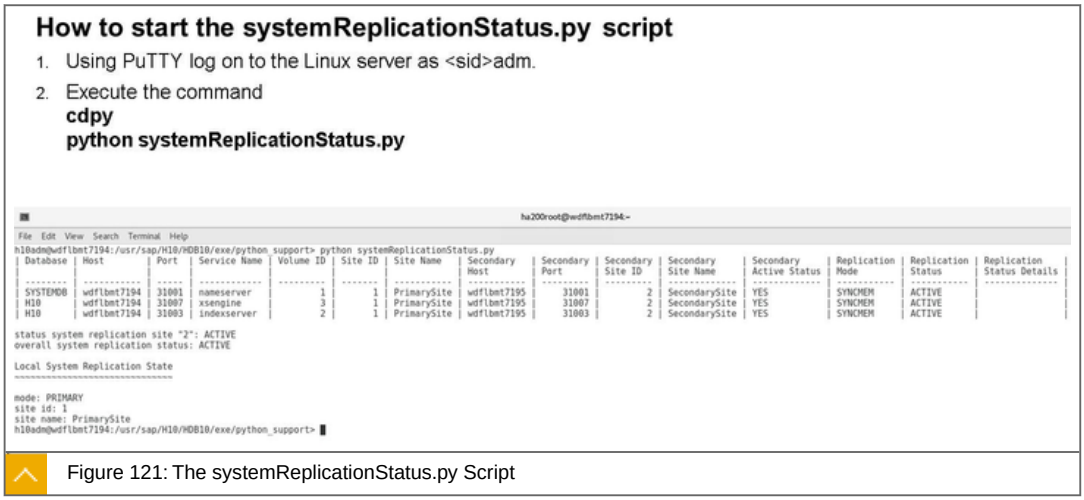


Figure 121: The systemReplicationStatus.py Script

During the exercises in Unit 4, we used the script systemReplicationStatus.py several times. This script is very useful to quickly check the overall status of the system replication setup using the command line.

You can monitor SAP HANA system replication using a command line script. Check the overall status of the system replication using the script systemReplicationStatus.py as the <sid>adm user.

The script provides the following return codes:

10: No System Replication

11: Error

12: Unknown

13: Initializing

14: Syncing

15: Active

Useful SAP Notes

SAP Note: 2518979 - HANA : how to check system replication status

LESSON SUMMARY
You should now be able to:

Understand the Python support scripts used in SAP HANA

Unit 5
Lesson 2

Appendix: Reinitializing a Non-Recoverable System Database

LESSON OBJECTIVES
After completing this lesson, you will be able to:

- Reinitialize a non-recoverable system database

Reinitialize a System Database

The system DB is corrupted and cannot startup anymore, but all tenants are still working. You need to recreate your system DB and do not have a backup.

In this situation it is possible to reinitialize the system DB instead of recovering the whole system with all tenants.

The command-line tool HDBMDCUTILallows you to reinitialize the system database with information on all tenant databases of the SAP HANA instance.

Caution:
This feature will not restore any other data previously stored in the system database. This data, for example license information, users, delivery units, security templates and others, has to be restored manually using the SQL interface or other user interfaces and/or tools.

Because of this limitation, the system database should always be recovered from a backup instead of using the HDBMDCUTILtool.

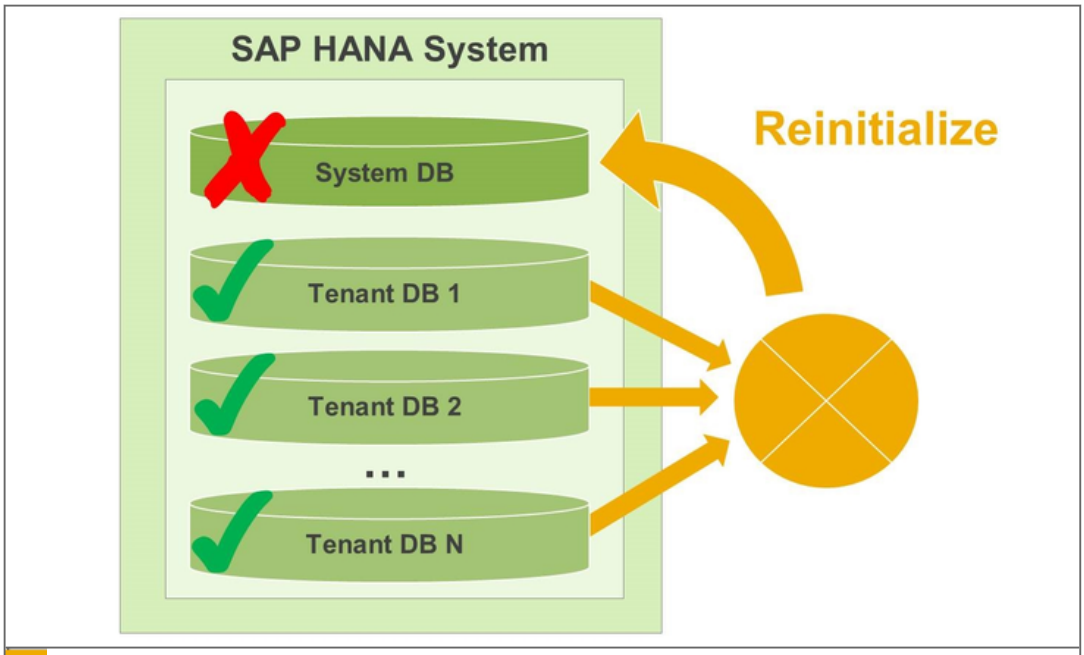


Figure 122: Reinitialize of a SystemDB

Reinitialize a System Database Using the Command-Line Tool hdbmdcutil

1. Stop the SAP HANA system. Ensure that all hosts and services that are part of this system have been stopped.
2. Start the HDBMDCUTIL tool with the -restoreTopology option: **hdbmdcutil -restoreTopology**

   To provide the password of the SYSTEM user of the system database, use the --set_user_system_pw option:

   hdbmdcutil -restoreTopology --set_user_system_pw
3. Enter the password of the SYSTEM user. Do not provide the password in the command line.
4. Start the SAP HANA system.

Additional Information	SAP Note
Re-initialize a not recoverable system database	2588284
How to reset SYSTEM password	1925267

Figure 123: Reinitialize System Database: Additional Information

LESSON SUMMARY
You should now be able to:

Reinitialize a non-recoverable system database

Learning Assessment

1. The HDBAdmin tool is the primary monitoring tool for SAP HANA, and is fully supported by SAP support.

   Determine whether this statement is true or false.

   - ☐ True
   - ☐ False

2. Which SAP HANA tool can be used to automatically check the availability of the SAP HANA database?

   Choose the correct answer.

   - ☐ **A** HDBAdmin
   - ☐ **B** HANAsitter
   - ☐ **C** HANAcleaner
   - ☐ **D** HDBscheduler

3. Which part of the data is restored when you reinitialize a non-recoverable system database?

   Choose the correct answer.

   - ☐ **A** License information
   - ☐ **B** Topology of the system database
   - ☐ **C** Delivery units
   - ☐ **D** Users and Roles

Learning Assessment - Answers

1. The HDBAdmin tool is the primary monitoring tool for SAP HANA, and is fully supported by SAP support.

   Determine whether this statement is true or false.

   [ ] True

   [X] False

   You are correct! The HDBAdmin tool is an internal SAP support tool. The customer can use it, but it is **not** supported by SAP.

2. Which SAP HANA tool can be used to automatically check the availability of the SAP HANA database?

   Choose the correct answer.

   [ ] **A** HDBAdmin

   [X] **B** HANAsitter

   [ ] **C** HANAcleaner

   [ ] **D** HDBscheduler

   You are correct! HANAsitter can be used to automatically check the availability of the SAP HANA database.

3. Which part of the data is restored when you reinitialize a non-recoverable system database?

   Choose the correct answer.

   [ ] **A** License information

   [X] **B** Topology of the system database

   [ ] **C** Delivery units

   [ ] **D** Users and Roles

   You are correct! Reinitialization of the system database does not restore any data previously stored in the system database. This data, for example license information, users, delivery units, and security templates has to be restored manually.