SUSE

SUSE Best Practices

SAP

SAP HANA System Replication Scale-Up - Performance Optimized Scenario

with SAPHanaSR-angi

SUSE Linux Enterprise Server for SAP Applications 15

Fabian Herschel, Distinguished Architect SAP (SUSE)
Lars Pinne, System Engineer (SUSE)

SUSE

SAP HANA System Replication Scale-Up - Performance Optimized Scenario

with SAPHanaSR-angi

Date: 2025-03-06

SUSE® Linux Enterprise Server for SAP Applications is optimized in various ways for SAP* applications. This guide provides detailed information about installing and customizing SUSE Linux Enterprise Server for SAP Applications for SAP HANA system replication in the performance optimized scenario. The document focuses on the steps to integrate an already installed and working SAP HANA with system replication. It is based on SUSE Linux Enterprise Server for SAP Applications 15 SP6. The concept however can be used with SUSE Linux Enterprise Server for SAP Applications 15 SP4 or newer.

Contents

1 About this guide

1.1 Introduction

SUSE® Linux Enterprise Server for SAP Applications is optimized in various ways for SAP* applications. This guide provides detailed information about installing and customizing **SUSE Linux Enterprise Server for SAP Applications for SAP HANA system replication in the performance optimized scenario**.

“SAP customers invest in SAP HANA” is the conclusion reached by a recent market study carried out by Pierre Audoin Consultants (PAC). In Germany, half of the companies expect SAP HANA to become the dominant database platform in the SAP environment. Often the “SAP Business Suite* powered by SAP HANA*” scenario is already being discussed in concrete terms.

SUSE is accommodating this development by offering SUSE Linux Enterprise Server for SAP Applications, the recommended and supported operating system for SAP HANA. In close collaboration with SAP, cloud service and hardware partners, SUSE provides resource agents for customers to ensure the high availability of SAP HANA system replications.

1.1.1 Abstract

This guide describes planning, setup, and basic testing of SUSE Linux Enterprise Server for SAP Applications based on the high availability solution scenario "SAP HANA Scale-Up System Replication Performance Optimized".

From the application perspective, the following variants are covered:

- Plain system replication
- System replication with secondary site read-enabled
- Multi-tier (chained) system replication
- Multi-target system replication
- Multi-tenant database containers for all above

From the infrastructure perspective, the following variants are covered:

- 2-node cluster with disk-based SBD
- 3-node cluster with diskless SBD

- On-premises deployment on physical and virtual machines
- Public cloud deployment (usually needs additional documentation focusing on the cloud specific implementation details)

Deployment automation simplifies roll-out. There are several options available, particularly on public cloud platfoms. Ask your public cloud provider or your SUSE contact for more information.

See *Section 2, "Supported scenarios and prerequisites"* for details.

Note

In this guide the software package SAPHanaSR-angi is used. This package replaces the two packages SAPHanaSR and SAPHanaSR-ScaleOut. Thus new deployment should be done with SAPHanaSR-angi only. For upgrading existing clusters to SAPHanaSR-angi, please read the blog article https://www.suse.com/c/how-to-upgrade-to-saphanasr-angi/ .

1.1.2 Scale-up versus scale-out

The first set of scenarios includes the topology of *scale-up* solutions.

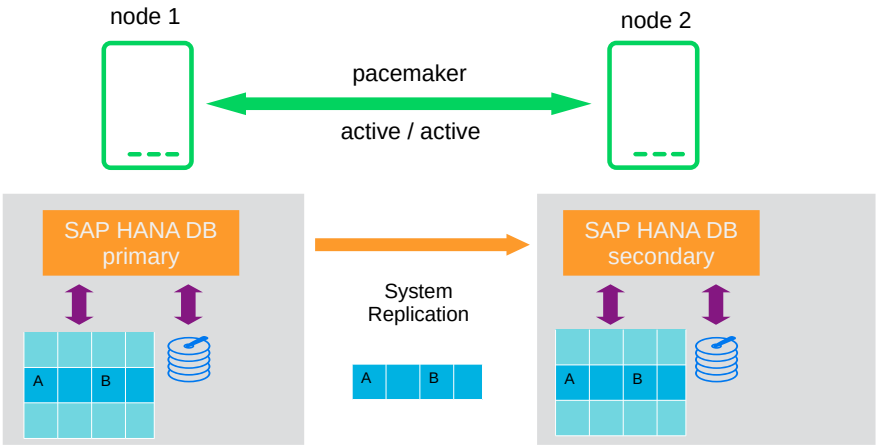


FIGURE 1: SAP HANA SYSTEM REPLICATION SCALE-UP IN THE CLUSTER

These scenarios are covered by the package SAPHanaSR-angi. System replication helps to replicate the database data from one computer to another computer to compensate for database failures (single-box replication).

The second set of scenarios includes the toplogy of *scale-out* solutions (multi-box replication). These scenarios are also covered by the package SAPHanaSR-angi.

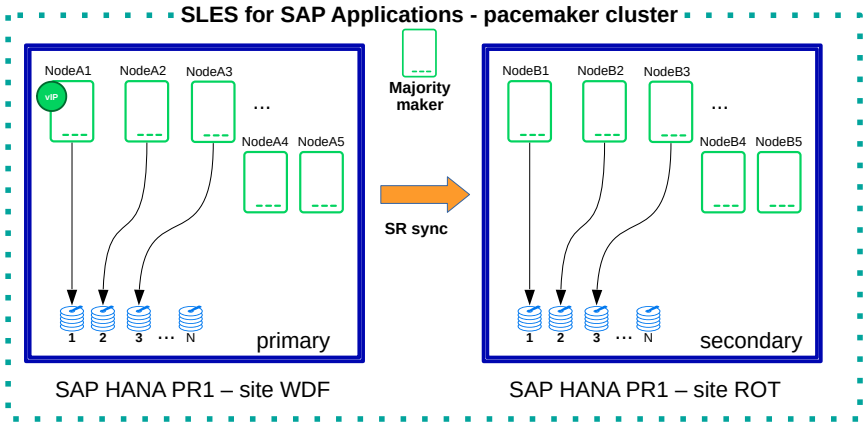


FIGURE 2: SAP HANA SYSTEM REPLICATION SCALE-OUT IN THE CLUSTER

With this mode of operation, internal SAP HANA high availability (HA) mechanisms and the resource agent must work together or be coordinated with each other. SAP HANA system replication automation for scale-out is described in a separate document available on our documentation Web page at https://documentation.suse.com/sbp/sap/. The document for scale-out is named *"SAP HANA System Replication Scale-Out - Performance Optimized Scenario"*.

1.1.3 Scale-up scenarios and resource agents

SUSE has implemented the scale-up scenario with the `SAPHanaController` resource agent (RA), which performs the actual check of the SAP HANA database instances. This RA is configured as a multi-state resource. In the scale-up scenario, the promoted RA instance assumes responsibility for the SAP HANA databases running in primary mode. The non-promoted RA instance is responsible for instances that are operated in synchronous (secondary) status.

To make configuring the cluster as simple as possible, SUSE has developed the `SAPHanaTopology` resource agent. This RA runs on all nodes of a SUSE Linux Enterprise Server for SAP Applications cluster and gathers information about the statuses and configurations of SAP HANA system replications. It is designed as a normal (stateless) clone.

SAP HANA system replication for scale-up is supported in the following scenarios or use cases:

- **Performance optimized** ($A \Rightarrow B$). This scenario and setup **is described in this document.**

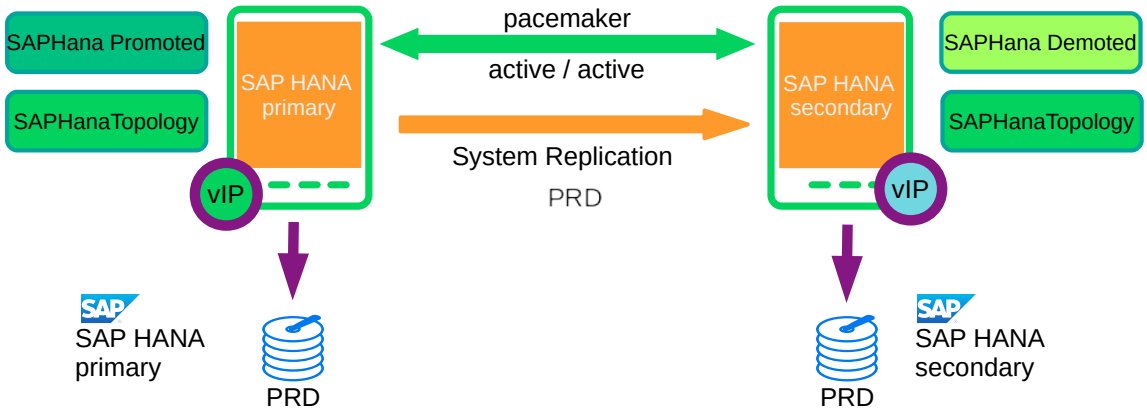


FIGURE 3: SAP HANA SYSTEM REPLICATION SCALE-UP IN THE CLUSTER - PERFORMANCE OPTIMIZED

In the performance optimized scenario an SAP HANA RDBMS site A is synchronizing with an SAP HANA RDBMS site B on a second node. As the SAP HANA RDBMS on the second node is configured to pre-load the tables, the takeover time is typically very short.
One big advance of the performance optimized scenario of SAP HANA is the possibility to allow read access on the secondary database site. To support this **read enabled** scenario, a second virtual IP address is added to the cluster and bound to the secondary role of the system replication.

- **Cost optimized** (*A ⇒ B, Q*). This scenario and setup is described in another document available from the documentation Web page (https://documentation.suse.com/sbp/sap/). The document for *cost optimized* is named *"SAP HANA System Replication Scale-Up - Cost Optimized Scenario"*.

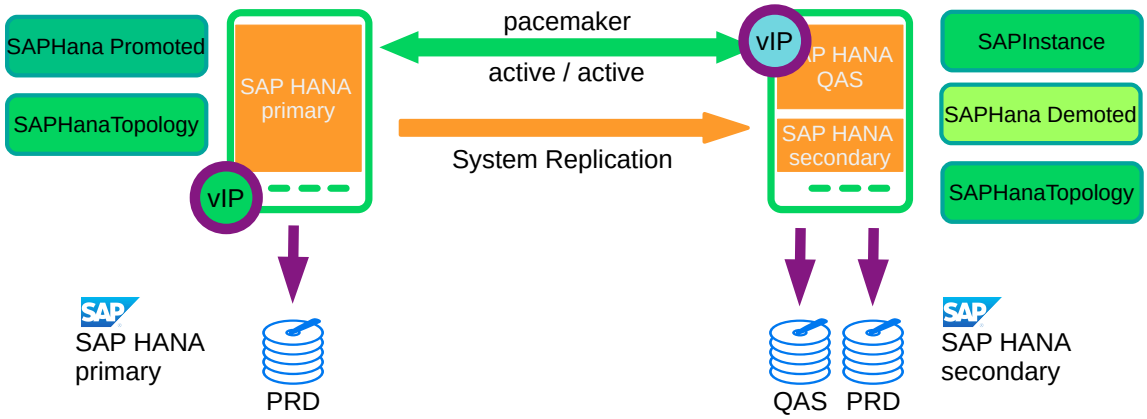


FIGURE 4: SAP HANA SYSTEM REPLICATION SCALE-UP IN THE CLUSTER - COST OPTIMIZED

In the cost optimized scenario, the second node is also used for a stand-alone non-replicated SAP HANA RDBMS system (like QAS or TST). Whenever a takeover is needed, the non-replicated system must be stopped first. As the productive secondary system on this node must be limited in using system resources, the table preload must be switched off. A possible takeover needs longer than in the performance optimized use case.
In the cost optimized scenario, the secondary needs to be running in a reduced memory consumption configuration. This is why *read enabled* must not be used in this scenario.
As already explained, the secondary SAP HANA database must run with memory resource restrictions. The HA/DR provider needs to remove these memory restrictions when a takeover occurs. This is why multi SID (also MCOS) must not be used in this scenario.

- **Multi-tier** ([A ⇒ B] → C) and **Multi-target** ([B ⇐ A] → C).

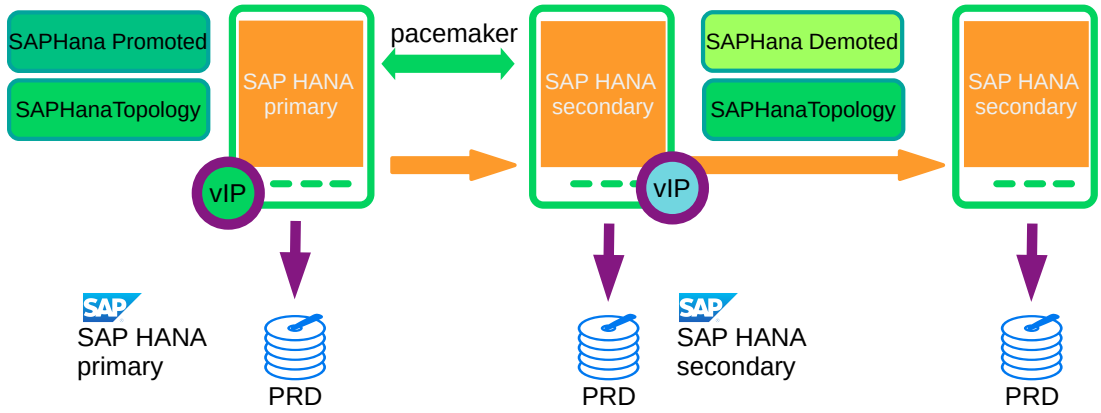


FIGURE 5: SAP HANA SYSTEM REPLICATION SCALE-UP IN THE CLUSTER - PERFORMANCE OPTIMIZED CHAIN

A *multi-tier* system replication has an additional target. In the past, this third side must have been connected to the secondary (chain topology). With current SAP HANA versions, the *multiple target topology* is allowed by SAP. Have a look at the scenarios and prerequisites section below or consult the manual pages SAPHanaSR(7) and susHanaSR.py(7) for details.

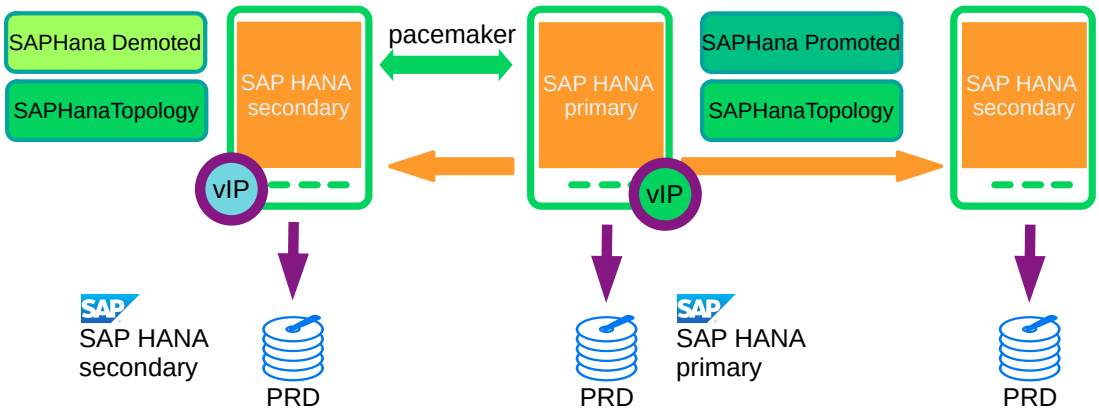


FIGURE 6: SAP HANA SYSTEM REPLICATION SCALE-UP IN THE CLUSTER - PERFORMANCE OPTIMIZED MULTI-TARGET

Multi-tier and multi-target systems are implemented as described in this document. Only the first replication pair (A and B) is handled by the cluster itself.

- **Multi-tenancy** or MDC.
  Multi-tenancy is supported for all above scenarios and use cases. This scenario is supported since SAP HANA SPS09. The setup and configuration from a cluster point of view is the same for multi-tenancy and single container. Thus you can use the above documents for both kinds of scenarios.

1.1.4 The concept of the performance optimized scenario

In case of failure of the primary SAP HANA on node 1 (node or database instance) the cluster first tries to start the takeover process. This allows to use the already loaded data at the secondary site. Typically, the takeover is much faster than the local restart.

To achieve an automation of this resource handling process, you must use the SAP HANA resource agents included in SAPHanaSR-angi. System replication of the productive database is automated with SAPHanaController and SAPHanaTopology.

The cluster only allows a takeover to the secondary site if the SAP HANA system replication was in sync until the point when the service of the primary got lost. This ensures that the last commits processed on the primary site are already available at the secondary site.

SAP did improve the interfaces between SAP HANA and external software, such as cluster frameworks. These improvements also include the implementation of SAP HANA call outs in case of special events, such as status changes for services or system replication channels. These call

outs are also called HA/DR providers. These interfaces can be used by implementing SAP HANA hooks written in python. SUSE has enhanced the SAPHanaSR package to include such SAP HANA hooks to optimize the cluster interface. Using the SAP HANA hooks described in this document allows to inform the cluster immediately if the SAP HANA system replication is broken. In addition to the SAP HANA hook status, the cluster continues to poll the system replication status on a regular basis.

You can adjust the level of automation by setting the parameter `AUTOMATED_REGISTER`. If automated registration is activated, the cluster will automatically register a former failed primary to become the new secondary. Refer to the manual pages SAPHanaSR(7) and ocf_suse_SAPHana(7) for details on all supported parameters and features.

Important

The solution is not designed to manually 'migrate' the primary or secondary instance using HAWK or any other cluster client commands. In the *Administration* section of this document we describe how to 'migrate' the primary to the secondary site using SAP and cluster commands.

1.2 Ecosystem of the document

1.2.1 Additional documentation and resources

Chapters in this manual contain links to additional documentation resources that are either available on the system or on the Internet.

For the latest documentation updates, see https://documentation.suse.com/.

You can find numerous whitepapers, best practices, setup guides, and other resources on the SUSE Linux Enterprise Server for SAP Applications best practices Web page: https://documentation.suse.com/sbp/sap/. In particular, there is an overview of all SUSE high availability solutions for SAP HANA and SAP S/4HANA workloads. Find the overview of high availability solutions supported by SUSE Linux Enterprise Server for SAP Applications here:

https://documentation.suse.com/sles-sap/sap-ha-support/html/sap-ha-support/article-sap-ha-support.html

SUSE also publishes blog articles about SAP and high availability. Join us by using the hashtag #TowardsZeroDowntime. Use the following link: https://www.suse.com/c/tag/TowardsZeroDowntime/.

Finally, there are manual pages shipped with the product.

1.2.2 Errata

To deliver urgent smaller fixes and important information in a timely manner, the Technical Information Document (TID) for this setup guide will be updated, maintained and published at a higher frequency:

- SAP HANA SR Performance Optimized Scenario - Setup Guide - Errata https://www.suse.com/support/kb/doc/?id=7023882
- Showing SOK Status in Cluster Monitoring Tools Workaround https://www.suse.com/support/kb/doc/?id=7023526 - see also the blog article https://www.suse.com/c/lets-flip-the-flags-is-my-sap-hana-database-in-sync-or-not/

1.2.3 Feedback

Several feedback channels are available:

Bugs and Enhancement Requests

For services and support options available for your product, refer to http://www.suse.com/support/.

To report bugs for a product component, go to https://scc.suse.com/support/ requests, log in, and select *Submit New SR* (Service Request).

Mail

For feedback on the documentation of this product, you can send a mail to doc-team@suse.com (mailto:doc-team@suse.com). Make sure to include the document title, the product version and the publication date of the documentation. To report errors or suggest enhancements, provide a concise description of the problem and refer to the respective section number and page (or URL).

2 Supported scenarios and prerequisites

For the `SAPHanaSR-angi` package configuration as decribed in this document, we limit the support to scale-up (single-box to single-box) system replication with the following configurations and parameters:

- Two-node clusters are standard. Three node clusters are fine if you install the resource agents also on that third node. But define in the cluster that SAP HANA resources must never run on that third node. In this case the third node is an additional majority maker in case of cluster separation.
- The cluster must include a valid STONITH method.
  - Any STONITH mechanism supported for production use by SUSE Linux Enterprise High Availability 15 (like SBD, IPMI) is supported with SAPHanaSR-angi.
  - This guide is focusing on the SBD fencing method as this is hardware independent.
  - If you use disk-based SBD as the fencing mechanism, you need one or more shared drives. For productive environments, we recommend more than one SBD device. For details on disk-based SBD, read the product documentation for SUSE Linux Enterprise High Availability and the manual pages sbd(8) and stonith_sbd(7).
  - For diskless SBD, you need at least three cluster nodes. The diskless SBD mechanism has the benefit that you do not need a shared drive for fencing. Since diskless SBD is based on self-fencing, reliable detection of lost quorum is absolutely crucial.
  - Priority fencing is an optional improvement for two nodes, but does not work for three nodes.
- Both nodes are in the same network segment (layer 2). Similar methods provided by cloud environments such as overlay IP addresses and load balancer functionality are also fine. Follow the cloud specific guides to set up your SUSE Linux Enterprise Server for SAP Applications cluster.
- Technical users and groups, such as *<sid>adm* are defined locally in the Linux system. If that is not possible, additional measures are needed to ensure reliable resolution of users, groups and permissions at any time. This might include caching.

- Name resolution of the cluster nodes and the virtual IP address must be done locally on all cluster nodes. If that is not possible, additional measures are needed to ensure reliable resolution of host names at any time.
- Time synchronization between the cluster nodes, such as NTP, is required.
- Both SAP HANA instances of the system replication pair (primary and secondary) have the same SAP Identifier (SID) and instance number.
- If the cluster nodes are installed in different data centers or data center areas, the environment must match the requirements of the SUSE Linux Enterprise High Availability cluster product. Of particular concern are the network latency and recommended maximum distance between the nodes. Review the product documentation for SUSE Linux Enterprise High Availability about those recommendations.
- Automated registration of a failed primary after takeover prerequisites need to be defined.
    - As a good starting configuration for projects, we recommend to switch off the automated registration of a failed primary. The setup `AUTOMATED_REGISTER="false"` is set as default. In this case, you need to register a failed primary after a takeover manually. For re-registration, use precisely the site names that are already known by the cluster. Use SAP tools like SAP HANA cockpit or *hdbnsutil*.
    - For optimal automation, we recommend to set `AUTOMATED_REGISTER="true"`.
- Automated start of SAP HANA instances during system boot must be switched off.
- Multi-tenancy (MDC) databases are supported.
    - Multi-tenancy databases can be used in combination with any other setup (performance-optimized, cost-optimized, multi-tier, multi-target and read-enabled).
    - In MDC configurations, the SAP HANA RDBMS is treated as a single system including all database containers. Therefore, cluster takeover decisions are based on the complete RDBMS status independent of the status of individual database containers.
    - Tests on Multi-tenancy databases can force a different test procedure if you are using strong separation of the tenants. As an example, killing the complete SAP HANA instance using *HDB kill* does not work, because the tenants are running with different Linux user UIDs. *<sid>adm* is not allowed to terminate the processes of the other tenant users.

- Only one system replication between the two SAP HANA database in the Linux cluster. Maximum one system replication to an SAP HANA database outside the Linux cluster.
    - Once an SAP HANA system replication site is known to the Linux cluster, that exact site name needs to be used whenever the site is registered manually.
    - If a third SAP HANA site is connected by system replication, that SAP HANA is not controlled by another Linux cluster. If that third site should work as part of a fall-back HA cluster in DR case, that HA cluster needs to be in standby.
    - The replication mode is either sync or syncmem for the controlled replication. Replication mode async is not supported. The operation modes delta_datashipping, logreplay and logreplay_readaccess are supported.
    - See also the dedicated section on requirements for susHanaSR.py.
- The current resource agent supports SAP HANA in system replication beginning with SAP HANA version 2.0 SPS05 revision 59.04. Even in SAP HANA multi-target environments, the current resource agent manages only two sites. Thus only two SAP HANA sites are part of the Linux cluster.
- Besides SAP HANA you need SAP hostagent installed and started on your system.
    - For SystemV style, the *sapinit* script needs to be active.
    - For systemd style, the service SAP<SID>_<INO> can stay enabled. The systemd enabled saphostagent and instance´s sapstartsrv is supported. Refer to the OS documentation for the systemd version. SAP HANA comes with native systemd integration as default starting with version 2.0 SPS07. Refer to SAP documentation for information on other SAP HANA versions.
    - Combining systemd style hostagent with SystemV style instance is allowed. However, all nodes in one Linux cluster need to use the same style.
- The RA's monitoring operations need to be active.
- Using HA/DR provider hook for srConnectionChanged() by enabling susHanaSR.py is mandatory.
- RA and HA/DR provider hook script´s runtime almost completely depends on call-outs to controlled resources, OS and Linux cluster. The infrastructure needs to allow these call-outs to return in time.

- Colocation constraints between the SAPHanaController RA and other resources are allowed only if they do not affect the RA's scoring. The location scoring finally depends on system replication status and must not be overruled by additional constraints. Thus it is not allowed to define rules forcing an SAPHanaController promoted clone to follow another resource.
- Reliable access to the /hana/shared/ file system is crucial for SAP HANA and the Linux cluster.
- SAP HANA feature Secondary Time Travel is not supported.
- The SAP HANA Fast Restart feature on RAM-tmfps and SAP HANA on persistent memory can be used, as long as they are transparent to Linux HA.
- No manual actions must be performed on the SAP HANA database while it is controlled by the Linux cluster. All administrative actions need to be aligned with the cluster.

For the HA/DR provider hook scripts *susHanaSR.py* and *susTkOver.py*, the following requirements apply:

- SAP HANA 2.0 SPS05 revision 059.04 and later provides Python3 and the HA/DR provider hook method **srConnectionChanegd()** with multi-target-aware parameters. Python 3 and multi-target-aware parameters are needed for the `SAPHanaSR-angi` package.
- SAP HANA 2.0 SPS05 and later provides the HA/DR provider hook method **preTakeover()**.
- The user *<sid>adm* needs execution permission as user root for the command *crm_attribute*.
- The hook provider needs to be added to the SAP HANA global configuration, in memory and on disk (in persistence).

For the HA/DR provider hook script *susChkSrv.py*, the following requirements apply:

- SAP HANA 2.0 SPS05 or later provides the HA/DR provider hook method srServiceStateChanged() with needed parameters.
- No other HA/DR provider hook script should be configured for the **srServiceStateChanged()** method. Hook scripts for other methods, provided in SAPHanaSR can be used in parallel to *susChkSrv.py*, if not documented contradictingly.
- The user <sid>adm needs execution permission as user root for the command SAPHanaSR-hookHelper.

- The hook provider needs to be added to the SAP HANA global configuration, in memory and on disk (in persistence).
- The hook script runs in the SAP HANA name server. It runs on the node where the event **srServiceStateChanged()** occurs.
- If susChkSrv.py parameter action_on_lost = stop is set and the RA SAPHana parameter AUTOMATED_REGISTER = true is set, it depends on HANA to release all OS resources prior to the registering attempt.
- If the hook provider should be pre-compiled, the particular Python version that comes with SAP HANA needs to be used.

See also manual pages SAPHanaSR(7), susHanaSR.py(7), susTkOver.py(7) and susChkSrv.py(7) for more details and requirements.

Important

Without a valid STONITH method, the complete cluster is unsupported and will not work properly.

If you need to implement a different scenario, we strongly recommend to define a Proof of Concept (PoC) with SUSE. This PoC will focus on testing the existing solution in your scenario. Most of the above mentioned limitations are set because careful testing is needed.

3 Scope of this document

This document describes how to set up the cluster to control SAP HANA in System Replication scenarios. The document focuses on the steps to integrate an already installed and working SAP HANA with System Replication. To create this document, SUSE Linux Enterprise Server for SAP Applications 15 SP6 was used. However, the concept can also be used with SUSE Linux Enterprise Server for SAP Applications 15 SP4 or newer.

The described example setup builds an SAP HANA HA cluster in two data centers in Walldorf (WDF) and in Rot (ROT), installed on two SLES for SAP 15 SP6 systems.

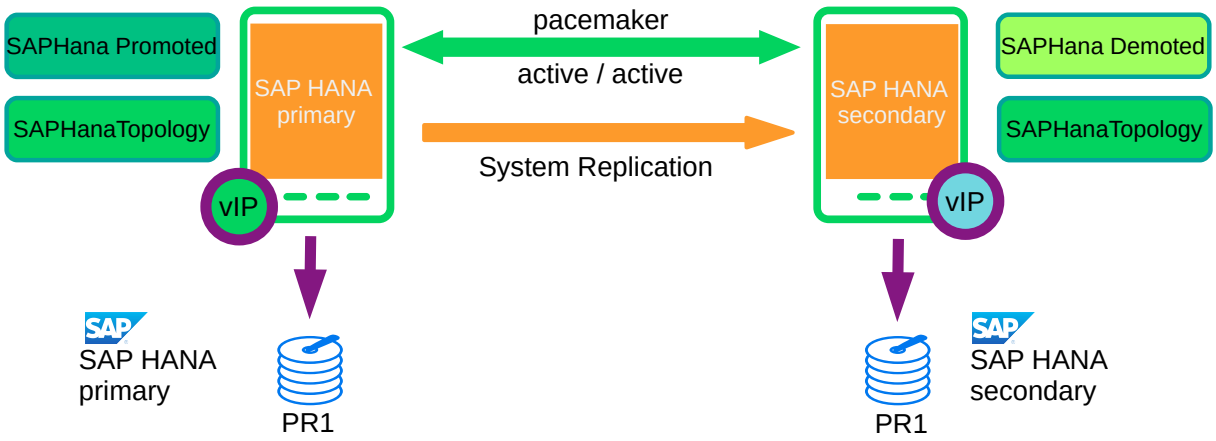


FIGURE 7: CLUSTER WITH SAP HANA SR - PERFORMANCE OPTIMIZED

You can either set up the cluster using the YaST wizard, doing it manually or using your own automation.

If you prefer to use the YaST wizard, you can use the shortcut *yast sap_ha* to start the module. The procedure to set up SAPHanaSR-angi using YaST is described in the product documentation of SUSE Linux Enterprise Server for SAP Applications in section *Setting Up an SAP HANA Cluster* at https://documentation.suse.com/sles-sap/15-SP6/html/SLES-SAP-guide/cha-cluster.html.

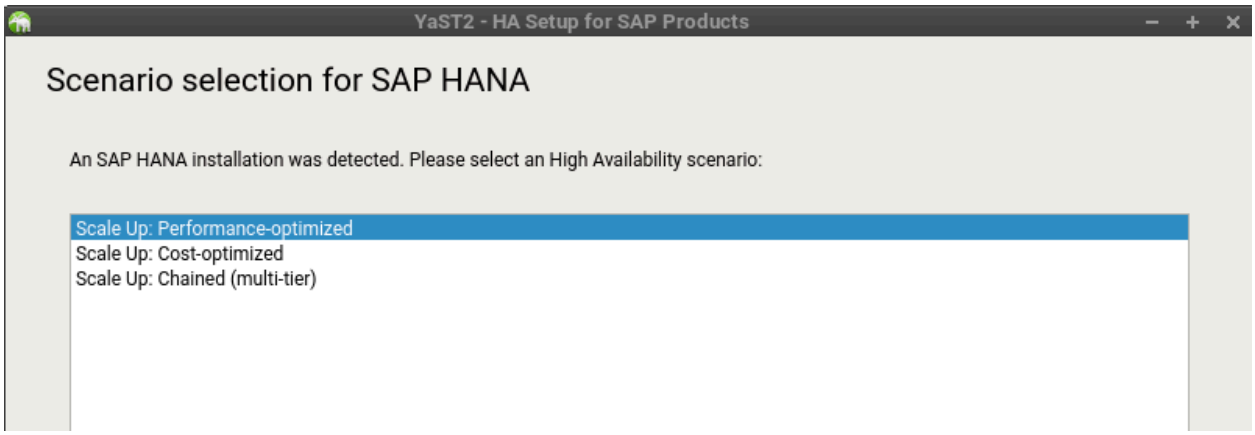
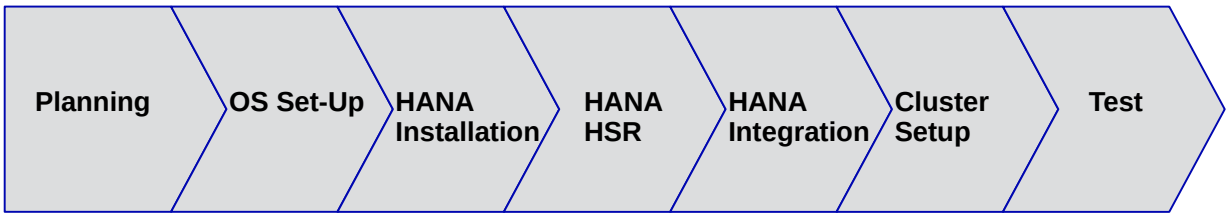


FIGURE 8: SCENARIO SELECTION FOR SAP HANA IN THE YAST MODULE SAP_HA

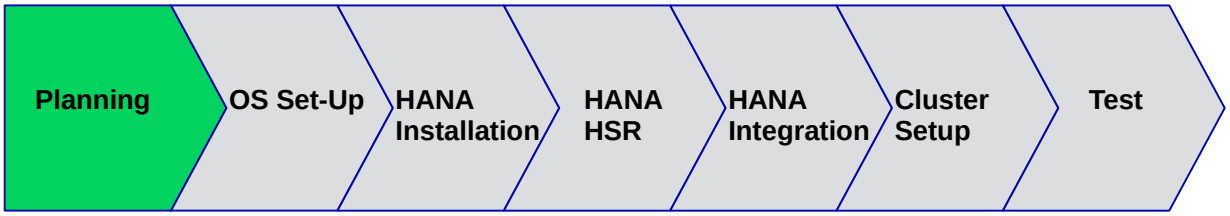
This guide focuses on the manual setup of the cluster to explain the details and to give you the possibility to create your own automation.

The seven main setup steps are:



- Planning (see *Section 4, "Planning the installation"*)
- OS installation (see *Section 5, "Setting up the operating system"*)
- Database installation (see *Section 6, "Installing the SAP HANA Databases on both cluster nodes"*)
- SAP HANA system replication setup (see *Section 7, "Setting up SAP HANA System Replication"*
- SAP HANA HA/DR provider hooks (see *Section 8, "Setting up SAP HANA HA/DR providers"*)
- Cluster configuration (see *Section 9, "Configuring the cluster"*)
- Testing (see *Section 10, "Testing the cluster"*)

4 Planning the installation



Planning the installation is essential for a successful SAP HANA cluster setup.

Before you start, you need the following:

- Software from SUSE: SUSE Linux Enterprise Server for SAP Applications installation media, a valid subscription, and access to update channels
- Software from SAP: SAP HANA installation media
- Physical or virtual systems including disks
- Filled parameter sheet (see below *Section 4.2, "Parameter sheet"* (page 20))

4.1 Minimum lab requirements and prerequisites

Note

The minimum lab requirements mentioned here are by no means SAP sizing information. These data are provided only to rebuild the described cluster in a lab for test purposes. Even for tests the requirements can increase, depending on your test scenario. For productive systems ask your hardware vendor or use the official SAP sizing tools and services.

Note

Refer to SAP HANA TDI documentation for allowed storage configuration and file systems.

Requirements with 1 SAP system replication instance per site (1 : 1) - without a majority maker (2 node cluster):

- 2 VMs with each 32GB RAM, 50GB disk space for the system
- 1 shared disk for SBD with 10 MB disk space
- 2 data disks (one per site) with a capacity of each 96GB for SAP HANA
- 1 additional IP address for takeover
- 1 optional IP address for the read-enabled setup
- 1 optional IP address for HAWK Administration GUI

Requirements with 1 SAP instance per site (1 : 1) - with a majority maker (3 node cluster):

- 2 VMs with each 32GB RAM, 50GB disk space for the system
- 1 VM with 2GB RAM, 50GB disk space for the system
- 2 data disks (one per site) with a capacity of each 96GB for SAP HANA
- 1 additional IP address for takeover
- 1 optional IP address for the read-enabled setup
- 1 optional IP address for HAWK Administration GUI

4.2 Parameter sheet

Even if the setup of the cluster organizing two SAP HANA sites is quite simple, the installation should be planned properly. You should have all needed parameters like SID, IP addresses and much more in place. It is good practice to first fill out the parameter sheet and then begin with the installation.

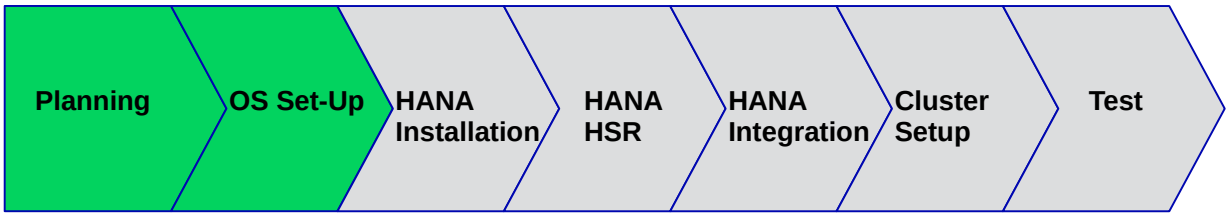
TABLE 1: PARAMETER SHEET FOR PLANNING

Parameter	Value	Role
Node 1		Cluster node name and IP address.
Node 2		Cluster node name and IP address.
Site A		Site name of the primary replicating SAP HANA database
Site B		Site name of the secondary replicating and the non-replicating SAP HANA database
SID		SAP System Identifier
Instance Number		Number of the SAP HANA database. For system replication also Instance Number + 1 is blocked.
Network mask		
vIP primary		Virtual IP address to be assigned to the primary SAP HANA site
vIP secondary		Virtual IP address to be assigned to the read-enabled secondary SAP HANA site (optional)
Storage		Storage for HDB data and log files is connected “locally” (per node; not shared)
SBD		STONITH device (two for production) or diskless SBD
HAWK Port	`7630`	
NTP Server		Address or name of your time server

TABLE 2: PARAMETER SHEET WITH VALUES USED IN THIS DOCUMENT

Parameter	Value	Role
Node 1	`suse01`, `192.168.1.11`	Cluster node name and IP address.
Node 2	`suse02`, `192.168.1.12`	Cluster node name and IP address.
SID	`HA1`	SAP System Identifier
Instance Number	`10`	Instance number of the SAP HANA database. For system replication also Instance Number + 1 is blocked.
Network mask	`255.255.255.0`	
vIP primary	`192.168.1.20`	
vIP secondary	`192.168.1.21`	(optional)
Storage		Storage for HDB data and log files is connected "locally" (per node; not shared)
SBD	`/dev/disk/by-id/SBDA`	STONITH device (three for production) or diskless
HAWK Port	`7630`	
NTP Server	pool pool.ntp.org	Address or name of your time server

5 Setting up the operating system



This section contains information you should consider during the installation of the operating system.

For the scope of this document, first SUSE Linux Enterprise Server for SAP Applications is installed and configured. Then the SAP HANA database including the system replication is set up. Finally the automation with the cluster is set up and configured.

5.1 Installing SUSE Linux Enterprise Server for SAP Applications

Multiple installation guides already exist, for different purposes and with different reasons to set up the server in a certain way. Below it is outlined where this information can be found. In addition, you will find important details you should consider to get a well-working system in place.

5.1.1 Installing the base operating system

Depending on your infrastructure and the hardware used, you need to adapt the installation. All supported installation methods and minimum requirement are described in the *Deployment Guide* for SUSE Linux Enterprise Server (https://documentation.suse.com/sles/15-SP6/single-html/SLES-deployment/). In case of automated installations you can find further information in the *AutoYaST Guide* (https://documentation.suse.com/sles/15-SP6/html/SLES-all/book-autoyast.html). The main installation guides for SUSE Linux Enterprise Server for SAP Applications that fit all requirements for SAP HANA are available from the SAP notes:

- 2578899 SUSE Linux Enterprise Server 15: Installation Note
- 2684254 SAP HANA DB: Recommended OS settings for SLES 15 / SLES for SAP Applications 15

5.1.2 Installing additional software

With SUSE Linux Enterprise Server for SAP Applications, SUSE delivers special resource agents for SAP HANA. With the pattern *sap-hana*, the old-style resource agent package `SAPHanaSR` is installed. This package needs to be replaced by the new `SAPHanaSR-angi` package. Follow the instructions below on each node if you have installed the systems based on SAP note 2578899. The pattern *High Availability* summarizes all tools recommended to be installed on **all** nodes, including the *majority maker*.

EXAMPLE 1: INSTALLING ADDITIONAL SOFTWARE FOR THE HA CLUSTER

1. Install the `High Availability` pattern. Do this on all nodes.

```
suse01:~ # zypper in --type pattern ha_sles
```

2. Uninstall the old-style package and install the new `SAPHanaSR-angi` resource agents. Do this on all nodes.

```
suse01:~ # rpm -e --nodeps SAPHanaSR SAPHanaSR-doc
suse01:~ # zypper in SAPHanaSR-angi
```

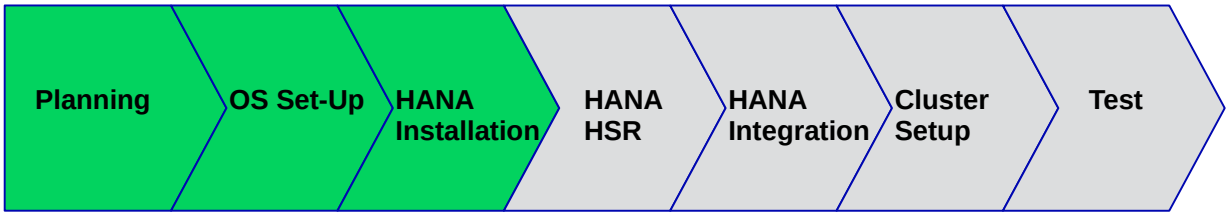
Note

Do not replace the package `SAPHanaSR` by `SAPHanaSR-angi` in an already running cluster. Upgrading from `SAPHanaSR` to `SAPHanaSR-angi` requires a specific procedure. See manual page SAPHanaSR_upgrade_to_angi(7) for details.

Installing the packages `supportutils-plugin-ha-sap` and `ClusterTools2` is highly recommended. The first helps collecting data for support requests, the second simplifies common administrative tasks.

For more information, see section *Installation and Setup* of the SUSE Linux Enterprise High Availability Administration Guide.

6 Installing the SAP HANA Databases on both cluster nodes



Even though this document focuses on the integration of an installed SAP HANA with system replication already set up into the Linux cluster, this chapter summarizes the test environment. Always use the official documentation from SAP to install SAP HANA and to set up the system replication.

This guide shows SAP HANA and saphostagent with native systemd integration. An example for legacy SystemV is outlined in the appendix *Section 12.5, "Example for checking legacy SystemV integration"*.

PROCEDURE

1. Install the SAP HANA databases.
2. Check if the SAP hostagent is installed on all cluster nodes. If this SAP service is not installed, install it now.
3. Verify that both databases are up and running.

6.1 Installing the SAP HANA databases

- Read the SAP Installation and Setup Manuals available at the SAP Marketplace.
- Download the SAP HANA Software from SAP Marketplace.
- Install the SAP HANA database as described in the SAP HANA Server Installation Guide. The SAP HANA database client will be installed together with the server by default.

6.2 Checking if the SAP hostagent is installed on all cluster nodes

Check if the native `systemd`-enabled SAP hostagent and instance `sapstartsrv` are installed on all cluster nodes. If not, install and enable them now.

As Linux user *root*, use the command `systemctl` and `systemd-cgls` to check the SAP hostagent and instance services:

```
# systemctl list-unit-files | grep sap
saphostagent.service enabled
sapinit.service generated
saprouter.service disabled
saptune.service enabled
```

The mandatory `saphostagent` service is enabled. This is the installation default. Some more SAP related services might be enabled, for example the recommended `saptune`.

```
# systemctl list-unit-files | grep SAP
SAPHA1_10.service enabled
```

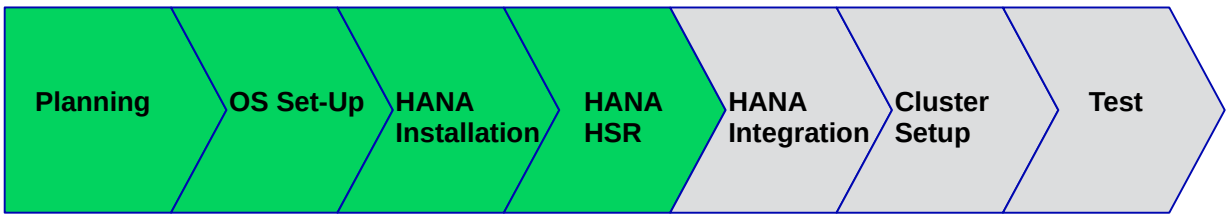
The instance service is indeed enabled, as required.

6.3 Verifying both databases are up and running

```
# systemd-cgls -u SAP.slice
Unit SAP.slice (/SAP.slice):
├─saphostagent.service
│ ├─2630 /usr/sap/hostctrl/exe/saphostexec pf=/usr/sap/hostctrl/exe/host_profile -systemd
│ ├─2671 /usr/sap/hostctrl/exe/sapstartsrv pf=/usr/sap/hostctrl/exe/host_profile -D
│ └─3591 /usr/sap/hostctrl/exe/saposcol -l -w60 pf=/usr/sap/hostctrl/exe/host_profile
└─SAPHA1_10.service
  ├─ 1257 hdbcompileserver
  ├─ 1274 hdbpreprocessor
  ├─ 1353 hdbindexserver -port 31003
  ├─ 1356 hdbxsengine -port 31007
  ├─ 2077 hdbwebdispatcher
  ├─ 2300 hdbrsutil --start --port 31003 --volume 3 --volumesuffix mnt00001/
hdb00003.00003 --identifier 1644426276
  ├─28462 /usr/sap/HA1/HDB10/exe/sapstartsrv pf=/usr/sap/HA1/SYS/profile/HA1_HDB10_suse01
  ├─31314 sapstart pf=/usr/sap/HA1/SYS/profile/HA1_HDB10_suse01
  ├─31372 /usr/sap/HA1/HDB10/suse01/trace/hdb.sapHA1_HDB10 -d -nw -f /usr/sap/HA1/HDB10/
suse01/daemon.ini pf=/usr/sap/HA1/SYS/profile/HA1_HDB10_suse01
  ├─31479 hdbnameserver
  └─32201 hdbrsutil --start --port 31001 --volume 1 --volumesuffix mnt00001/hdb00001 --
identifier 1644426203
```

The SAP hostagent `saphostagent.service` and the instance´s `sapstartsrv` `SAPHA1_10.service` are running in the `SAP.slice`. See also manual pages systemctl(8) and systemd-cgls(8) for details.

7 Setting up SAP HANA System Replication



For more information read the section *Setting Up System Replication* of the SAP HANA Administration Guide.

Procedure

1. Back up the primary database.
2. Enable the primary database.
3. Register and start the secondary database.
4. Verify the system replication.

7.1 Backing up the primary database

Back up the primary database as described in the SAP HANA Administration Guide, section *SAP HANA Database Backup and Recovery*. We provide an example with SQL commands. You need to adapt these backup commands to match your backup infrastructure.

EXAMPLE 2: SIMPLE BACKUP FOR THE SYSTEM DATABASE AND ALL TENANTS WITH ONE SINGLE BACKUP CALL

> As user *<sid>adm* enter the following command:
>
> ```
> ~> hdbsql -i 10 -u SYSTEM -d SYSTEMDB \
>    "BACKUP DATA FOR FULL SYSTEM USING FILE ('backup')"
> ```
>
> You will get a command output similar to the following:
>
> ```
> 0 rows affected (overall time 15.352069 sec; server time 15.347745 sec)
> ```

EXAMPLE 3: SIMPLE BACKUP FOR A SINGLE CONTAINER (NON MDC) DATABASE

> Enter the following command as user *<sid>adm*:
>
> ```
> ~> hdbsql -i <instanceNumber> -u <dbuser> \
>    "BACKUP DATA USING FILE ('backup')"
> ```

Important

Without a valid backup, you cannot bring SAP HANA into a system replication configuration.

7.2 Enabling the primary node

As Linux user *<sid>adm*, enable the system replication at the primary node. You need to define a site name (like WDF). This site name must be unique for all SAP HANA databases which are connected via system replication. This means the secondary must have a different site name. The site names must not be changed later when the cluster has been activated.

Note

Do not use strings like "primary" and "secondary" as site names.

EXAMPLE 4: ENABLE THE PRIMARY

Enable the primary using the `-sr_enable` option.

```
suse01:~> hdbnsutil -sr_enable --name=WDF
checking local nameserver:
checking for active nameserver ...
nameserver is running, proceeding ...
configuring ini files ...
successfully enabled system as primary site ...
done.
```

EXAMPLE 5: CHECK SR CONFIGURATION ON THE PRIMARY

Check the primary using the command `hdbnsutil -sr_stateConfiguration`.

```
suse01:~> hdbnsutil -sr_stateConfiguration --sapcontrol=1
SAPCONTROL-OK: <begin>
mode=primary
site id=1
site name=WDF
SAPCONTROL-OK: <end>
done.
```

The mode has changed from “none” to “primary”. The site now has a site name and a site ID.

7.3 Registering the secondary node

The SAP HANA database instance on the secondary side must be stopped before the instance can be registered for the system replication. You can use your preferred method to stop the instance (like `HDB` or `sapcontrol`). After the database instance has been stopped successfully, you can register the instance using `hdbnsutil`. Again, use the Linux user *<sid>adm*:

EXAMPLE 6: STOP THE SECONDARY

To stop the secondary, you can use the command line tool *HDB*.

```
suse02:~> HDB stop
```

EXAMPLE 7: COPY THE KEY AND KEY-DATA FILE FROM THE PRIMARY TO THE SECONDARY SITE

Beginning with SAP HANA 2.0, the system replication is running encrypted. The key files need to be copied-over from the primary to the secondary site.

```
~> cd /usr/sap/<SID>/SYS/global/security/rsecssfs
~> rsync -va {,<node1-siteB>:}$PWD/data/SSFS_<SID>.DAT
~> rsync -va {,<node1-siteB>:}$PWD/key/SSFS_<SID>.KEY
```

EXAMPLE 8: REGISTER THE SECONDARY

The registration of the secondary is triggered by calling *hdbnsutil -sr_register* ….

```
...
suse02:~> hdbnsutil -sr_register --name=ROT \
     --remoteHost=suse01 --remoteInstance=10 \
     --replicationMode=sync --operationMode=logreplay
adding site ...
checking for inactive nameserver ...
nameserver suse02:30001 not responding.
collecting information ...
updating local ini files ...
done.
```

The *remoteHost* is the primary node in our case, the *remoteInstance* is the database instance number (here 10).

Now start the database instance again and verify the system replication status. On the secondary node, the mode should be one of "SYNC" or "SYNCMEM". "ASYNC" is **not supported with automated cluster takeover**. The mode depends on the **replicationMode** option defined during the registration of the secondary.

EXAMPLE 9: START SECONDARY AND CHECK SR CONFIGURATION

To start the new secondary, use the command line tool *HDB*. Then check the SR configuration using `hdbnsutil -sr_stateConfiguration`.

```
suse02:~> HDB start
...
suse02:~> hdbnsutil -sr_stateConfiguration --sapcontrol=1
SAPCONTROL-OK: <begin>
mode=sync
site id=2
site name=ROT
active primary site=1
primary masters=suse01
SAPCONTROL-OK: <end>
done.
```

To view the replication state of the whole SAP HANA cluster, use the following command as *<sid>adm* user on the primary node:

EXAMPLE 10: CHECKING SYSTEM REPLICATION STATUS DETAILS

The python script *systemReplicationStatus.py* provides details about the current system replication.

```
suse01:~> HDBSettings.sh systemReplicationStatus.py --sapcontrol=1
...
site/2/SITE_NAME=ROT
site/2/SOURCE_SITE_ID=1
site/2/REPLICATION_MODE=SYNC
site/2/REPLICATION_STATUS=ACTIVE
overall_replication_status=ACTIVE
site/1/REPLICATION_MODE=PRIMARY
site/1/SITE_NAME=WDF
local_site_id=1
...
```

7.4 Manually testing the SAP HANA SR takeover

Before you integrate your SAP HANA system replication into the HA cluster, it is mandatory to do a manual takeover. Testing without the cluster helps to make sure that basic operation (takeover and registration) is working as expected.

- Stop SAP HANA on node 1.
- Takeover SAP HANA to node 2.
- Register node 1 as secondary.
- Start SAP HANA on node 1.
- Wait until sync state is active.

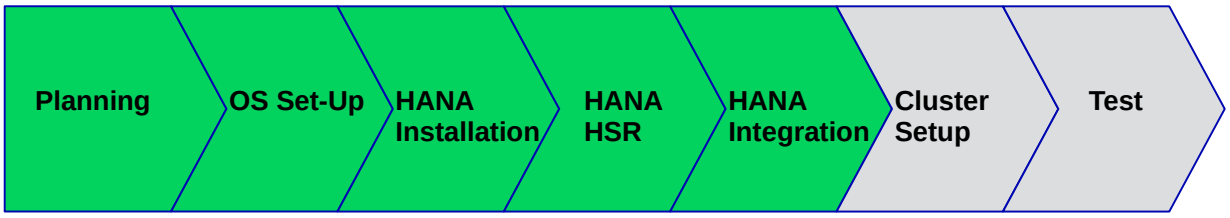
7.5 Optional: Manually re-establishing SAP HANA SR to original state

Bring the systems back to the original state:

- Stop SAP HANA on node 2.
- Take over SAP HANA to node 1.

- Register node 2 as secondary.
- Start SAP HANA on node2.
- Wait until sync state is active.

8 Setting up SAP HANA HA/DR providers



This step is mandatory to inform the cluster immediately if the secondary gets out of sync. The hook is called by SAP HANA using the HA/DR provider interface in point-of-time when the secondary gets out of sync. This is typically the case when the first commit pending is released. The hook is called by SAP HANA again when the system replication is back. This HA/DR provider method is **srConnectionChanged()**, the related SUSE hook script is *susHanaSR.py*. The hook script *susHanaSR.py* is defacto mandatory.

Another hook is called by SAP HANA before an SR takeover is processed. This method can be used to block a manual takeover during normal cluster operation. This HA/DR provider method is **preTakeover()**, the related SUSE hook script is *susTkOver.py*.

A third hook is called by SAP HANA when a service status changes. This method can be used to speed up the takeover in case the indexserver process fails. This HA/DR provider method is **srServiceStateChanged()**, the related SUSE hook script is *susChkSrv.py*.

Procedure

1. Implement the python hook script *susHanaSR.py* on both sites.
2. Implement the python hook script *susTkOver.py* on both sites.
3. Implement the python hook script *susChkSrv.py* on both sites.
4. Configure the system replication operation mode.
5. Allow *<sid>adm* to access the cluster.

6. Start SAP HANA.
7. Test the hook integration.

This will implement three SAP HANA HA/DR provider hook scripts. The hook script *susHanaSR.py* does not need any configuration parameters. The configuration for *susTkOver.py* normally does not need to be adapted. The default for parameter **sustkover_timeout** is set to 30 seconds, which is good for most environments. The configuration shown for *susChkSrv.py* is a good starting point. Any tuning should be aligned with the SAP experts.

Note

All hook scripts should be used directly from the `SAPHanaSR` package. If the scripts are moved or copied, regular SUSE package updates will not work.

SAP HANA must be stopped to change the *global.ini* and allow SAP HANA to integrate the HA/DR hook scripts during start. Alternatively, `SAPHanaSR-manageProvider` might be used for adapting the *global.ini*. See manual page SAPHanaSR-manageProvider(8) for details.

8.1 Implementing susHanaSR hook for srConnectionChanged

Use the hook from the SAPHanaSR-angi package */usr/share/SAPHanaSR-angi/susHanaSR.py*. The hook must be configured on all SAP HANA cluster nodes. In *global.ini*, the section *[ha_dr_provider_sushanasr]* needs to be created. The section *[trace]* might be adapted. Refer to the manual page susHanaSR.py(7) for details on this HA/DR provider hook script, see also SAPHanaSR-manageProvider(8).

EXAMPLE 11: STOP SAP HANA

Stop SAP HANA either with *HDB* or using *sapcontrol*.

```
~> sapcontrol -nr <instanceNumber> -function StopSystem
```

EXAMPLE 12: ADDING SAPHANASR VIA *GLOBAL.INI*

Best is to use the SAP HANA tools for changing *global.ini*. Alternatively you may use `SAPHanaSR-manageProvider`, see manual page SAPHanaSR-manageProvider(8).

```
[ha_dr_provider_sushanasr]
provider = susHanaSR
path = /usr/share/SAPHanaSR-angi/
execution_order = 1
```

```
[trace]
ha_dr_sushanasr = info
```

8.2 Implementing susTkOver hook for preTakeover

Use the hook from the SAPHanaSR-angi package */usr/share/SAPHanaSR-angi/susTkOver.py*. The hook must be configured on all SAP HANA cluster nodes. In *global.ini*, the section *[ha_dr_provider_sustkover]* needs to be created. The section *[trace]* might be adapted. Refer to the manual page susTkOver.py(7) for details on this HA/DR provider hook script, see also SAPHanaSR-manageProvider(8).

EXAMPLE 13: STOP SAP HANA

Stop SAP HANA either with *HDB* or using *sapcontrol*.

```
ha1adm@suse02:/usr/sap/HA1/HDB10> sapcontrol -nr <instanceNumber> -function
 StopSystem
```

EXAMPLE 14: ADDING *SUSTKOVER* VIA *GLOBAL.INI*

Best is to use the SAP HANA tools for changing *global.ini*. Alternatively you may use `SAPHanaSR-manageProvider`, see manual page SAPHanaSR-manageProvider(8).

```
[ha_dr_provider_sustkover]
provider = susTkOver
path = /usr/share/SAPHanaSR-angi/
execution_order = 2

[trace]
ha_dr_sustkover = info
...
```

8.3 Implementing susChkSrv hook for srServiceStateChanged

Use the hook from the SAPHanaSR-angi package */usr/share/SAPHanaSR-angi/susChkSrv.py*. The hook must be configured on all SAP HANA cluster nodes. In *global.ini*, the section *[ha_dr_provider_suschksrv]* needs to be created. The section *[trace]* might be adapted. Refer to the manual page susChkSrv.py(7) for details on this HA/DR provider hook script, see also SAPHanaSR-manageProvider(8).

EXAMPLE 15: STOP SAP HANA

Stop SAP HANA either with *HDB* or using *sapcontrol*.

```
ha1adm@suse02:/usr/sap/HA1/HDB10> sapcontrol -nr <instanceNumber> -function
 StopSystem
```

EXAMPLE 16: ADDING *SUSCHKSRV* VIA *GLOBAL.INI*

Best is to use the SAP HANA tools for changing *global.ini*. Alternatively you may use `SAPHanaSR-manageProvider`, see manual page SAPHanaSR-manageProvider(8).

```
[ha_dr_provider_suschksrv]
provider = susChkSrv
path = /usr/share/SAPHanaSR-angi/
execution_order = 3
action_on_lost=stop

[trace]
ha_dr_suschksrv = info
...
```

8.4 Configuring system replication operation mode

When your system is connected as SAP HANA system replication target, you can find an entry in the *global.ini* which defines the operation mode. Up to now there are the following modes available:

- *delta_datashipping*
- *logreplay*
- *logreplay_readaccess*

Until a takeover and re-registration in the opposite direction, the entry for the operation mode is missing on your primary site. The first operation mode which was available was *delta_datashipping*. Today the preferred modes for HA are *logreplay* or *logreplay_readaccess*. Using the operation mode *logreplay* makes your secondary site in the SAP HANA system replication a hot standby system. For more details regarding all operation modes, check the available SAP documentation such as the guide "How To Perform System Replication for SAP HANA " (https://www.sap.com/documents/2017/07/606a676e-c97c-0010-82c7-eda71af511fa.html).

EXAMPLE 17: CHECKING THE OPERATION MODE

Check both *global.ini* files and add the operation mode if needed. Check the section *system_replication* for entry ´operation_mode = logreplay´.

Path for the *global.ini*: /hana/shared/<SID>/global/hdb/custom/config/

```
[system_replication]
operation_mode = logreplay
```

8.5 Allowing <sid>adm to access the cluster

The current version of the susHanaSR python hook uses the command `sudo` to allow the *<sid>adm* user to access the cluster attributes.

The user *<sid>adm* must be able to set the cluster attributes **hana_<sid>_site_srHook_***. The SAP HANA system replication hook needs password free access. The following example limits the sudo access to exactly setting the needed attribute. The entries can be added to a new file */etc/sudoers.d/SAPHanaSR* so that the original */etc/sudoers* file does not need to be edited. See manual page sudoers(5) for details.

Replace the <sid> by the **lowercase** SAP system ID (like `ha1`).

- Entry in sudo permissions /etc/sudoers.d/SAPHanaSR file

Basic sudoers entry to allow <sid>adm to use the hooks SAPHanaSR and susTkOver.

```
# SAPHanaSR-ScaleUp entries for writing srHook cluster attribute and SAPHanaSR-hookHelper
<sid>adm ALL=(ALL) NOPASSWD: /usr/sbin/crm_attribute -n hana_<sid>_*
<sid>adm ALL=(ALL) NOPASSWD: /usr/bin/SAPHanaSR-hookHelper --sid=<SID> *
```

- More specific sudoers entries to meet a high security level

All Cmnd_Alias entries must be each defined as a single line entry. In our example, we have five separate lines with Cmnd_Alias entries and one line for the *<sid>adm* user permitting the Cmnd_Aliases. In the document at hand, however, the separate lines of the example might include a line-break forced by document formatting. The alias identifier (for example SOK_SITEA) needs to be in capitals.

Replace the <sid> by the **lowercase** SAP system ID (like `ha1`). Replace the <SID> by the **uppercase** SAP system ID.

```
# SAPHanaSR-ScaleUp entries for writing srHook cluster attribute
Cmnd_Alias SOK_SITEA    = /usr/sbin/crm_attribute -n hana_<sid>_site_srHook_<siteA> -v
 SOK   -t crm_config -s SAPHanaSR
Cmnd_Alias SFAIL_SITEA  = /usr/sbin/crm_attribute -n hana_<sid>_site_srHook_<siteA> -v
 SFAIL -t crm_config -s SAPHanaSR
Cmnd_Alias SOK_SITEB    = /usr/sbin/crm_attribute -n hana_<sid>_site_srHook_<siteB> -v
 SOK   -t crm_config -s SAPHanaSR
```

```
Cmnd_Alias SFAIL_SITEB  = /usr/sbin/crm_attribute -n hana_<sid>_site_srHook_<siteB> -v
 SFAIL -t crm_config -s SAPHanaSR
Cmnd_Alias HOOK_HELPER  = /usr/bin/SAPHanaSR-hookHelper --sid=<SID> --case=checkTakeover

<sid>adm ALL=(ALL) NOPASSWD: SOK_SITEA, SFAIL_SITEA, SOK_SITEB, SFAIL_SITEB, HOOK_HELPER
```

8.6 Starting SAP HANA

Start SAP HANA as user <sid>adm.

```
~> HDB start
```

8.7 Testing the hook integration

8.7.1 Check the load of the hook scripts

As user <sid>adm check the SAP HANA tracefiles, if the HA/DR provider scripts are loaded and called successfully during an SAP HANA event.

Check if script *susHanaSR.py* is loaded and initialized correctly:

```
~> cdtrace
~> grep HADR.*load.*susHanaSR nameserver_*.trc
~> grep susHanaSR.init nameserver_*.trc
```

Check if script *susTkOver.py* is loaded and initialized correctly:

```
~> cdtrace
~> grep HADR.*load.*susTkOver nameserver_*.trc
~> grep susTkOver.init nameserver_*.trc
```

Check if script *susChkSrv.py* is loaded and initialized correctly:

```
~> cdtrace
~> grep HADR.*load.*susChkSrv nameserver_*.trc
~> grep susChkSrv.init nameserver_*.trc
~> egrep '(LOST:|STOP:|START:|DOWN:|init|load|fail)' nameserver_suschksrv.trc
```

8.7.2 Check an **srConnectionChanged** event

After an event has been processed by the HA/DR provider script, check for the correct behaviour.

As user <sid>adm check the SAP HANA tracefiles and verify if *susHanaSR.py* did successfully interact with the cluster or created a fallback file.

```
~> cdtrace
~> grep susHanaSR.srConnection.*CRM nameserver_*.trc
~> grep susHanaSR.srConnection.*fallback nameserver_*.trc
```

8.7.3 Check a **preTakeover** event

After an event has been processed by the HA/DR provider script, check for the correct behaviour.

To test script *susTkOver.py* with stopped cluster, procedure *Section 7.4, "Manually testing the SAP HANA SR takeover"* (page 29) can be used here again. While the cluster is not set up at this moment the takeover will not be blocked.

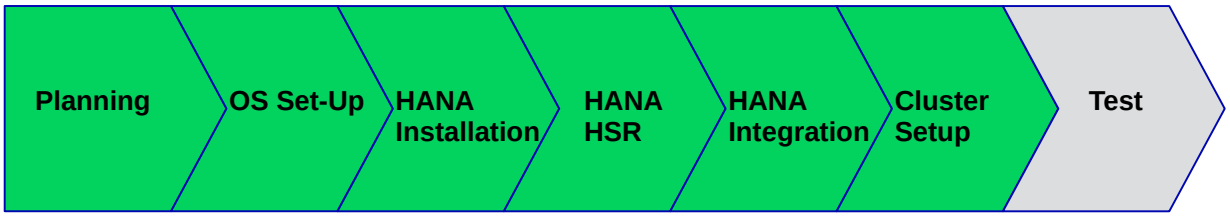
As user root check the system messages and verify if the sudo permissions for *susTkOver.py* calling SAPHanaSR-hookHelper are set successfully.

```
# grep "sudo.*SAPHanaSR-hookHelper"  /var/log/messages
```

As user <sid>adm check the SAP HANA tracefiles, to verify if *susTkOver.py* did successfully block or permit the takeover request. First check for permitted takeover requests, then check for blocked takeover requests.

```
~> cdtrace
~> grep susTkOver.preTakeover.*permit nameserver_*.trc
~> grep susTkOver.preTakeover.*failed.*50277 nameserver_*.trc
```

9 Configuring the cluster



This chapter describes the configuration of the cluster software SUSE Linux Enterprise High Availability, which is part of SUSE Linux Enterprise Server for SAP Applications, and the SAP HANA database integration.

ACTIONS

1. Basic cluster configuration
2. Configuration of cluster properties and resources
3. Testing the HA/DR provider hook integration

9.1 Configuring the basic cluster

The first step is to set up the basic cluster framework. For convenience, use YaST or the ha-cluster-init script. It is strongly recommended to add a second corosync ring, change it to UCAST communication and adjust the timeout values to fit your environment.

9.1.1 Setting up watchdog for "storage-based fencing"

If you use the storage-based fencing (SBD) mechanism (diskless or disk-based), you must also configure a watchdog. The watchdog is needed to reset a node if the system cannot longer access the SBD (diskless or disk-based). It is mandatory to configure the Linux system for loading a watchdog driver. It is strongly recommended to use a watchdog with hardware assistance (as is available on most modern systems), such as hpwdt, iTCO_wdt, or others. As fallback, you can use the softdog module.

EXAMPLE 18: SETUP FOR WATCHDOG

> **Important**
>
> Access to the watchdog timer: No other software must access the watchdog timer; it can only be accessed by one process at any time. Some hardware vendors ship systems management software that use the watchdog for system resets (for example HP ASR daemon). Such software must be disabled if the watchdog is to be used by SBD.

Determine the right watchdog module. Alternatively, you can find a list of installed drivers with your kernel version.

```
# ls -l /lib/modules/$(uname -r)/kernel/drivers/watchdog
```

Check if any watchdog module is already loaded.

```
# lsmod | egrep "(wd|dog|i6|iT|ibm)"
```

If you get a result, the system has already a loaded watchdog. If the watchdog does not match your watchdog device, you need to unload the module.

To safely unload the module, check first if an application is using the watchdog device.

```
# lsof /dev/watchdog
# rmmod <wrong_module>
```

Enable your watchdog module and make it persistent. For the example below, *softdog* has been used. However, *softdog* has some restrictions and should not be used as first option.

```
# echo softdog > /etc/modules-load.d/watchdog.conf
# systemctl restart systemd-modules-load
```

Check if the watchdog module is loaded correctly.

```
# lsmod | grep dog
# ls -l /dev/watchdog
```

Testing the watchdog can be done with a simple action. Ensure to switch of your SAP HANA first because the watchdog will force an unclean reset or shutdown of your system.

In case a hardware watchdog is used, a desired action is predefined after the timeout of the watchdog has reached. If your watchdog module is loaded and not controlled by any other application, do the following:

Important

Triggering the watchdog without continuously updating the watchdog resets/switches off the system. This is the intended mechanism. The following commands will force your system to be reset/switched off.

In case the softdog module is used, the following action can be performed:

```
# sync; cat /dev/watchdog & while date; do sleep 10; done
```

After your test was successful, you must implement the watchdog on all cluster members.

9.1.2 Setting up the initial cluster using `ha-cluster-init`

For more detailed information about setting up a cluster, refer to the sections *Setting Up the First Node* and *Adding the Second Node* of the Installation and Setup Quick Start for SUSE Linux Enterprise High Availability 15 SP6 at https://documentation.suse.com/sle-ha/15-SP6/single-html/SLE-HA-installation/.

This setup uses unicast (UCAST) for corosync communication (*-u* option). Refer to the https://documentation.suse.com/sle-ha/15-SP6/single-html/SLE-HA-administration/ on detailed explanations of the terms unicast/multicast.

Create an initial setup, using the `ha-cluster-init` command, and follow the dialogs. Do this only on the first cluster node. Answer "no" to "Do you wish to configure a virtual IP address" and to "Do you want to configure QDevice".

To use two corosync rings make sure you have two interfaces configured and run:

```
suse01:~ # ha-cluster-init -u -M -s /dev/disk/by-id/SBDA -s /dev/disk/by-id/SBDB
```

To use only one corosync ring leave out the *-M* option (not recommended):

```
suse01:~ # ha-cluster-init -u -s /dev/disk/by-id/SBDA -s /dev/disk/by-id/SBDB
```

This command configures the basic cluster framework including:

- SSH keys
- csync2 to transfer configuration files
- SBD (two in this guide, better three for production)
- corosync (at least one ring, better two rings)
- HAWK Web interface

Important

As requested by `ha-cluster-init`, change the password of the user hacluster.

9.1.3 Checking and adapting the corosync and SBD configuration

9.1.3.1 Checking the corosync configuration

Check the following blocks in the file */etc/corosync/corosync.conf*. The important parts are *udpu* and the correct ring/IP configuration.

See also the example at the end of this document and refer to the manual pages corosync.conf(5), votequorum(5) and corosync_overview(8) for details on parameters and features.

```
totem {
    ...

    interface {
        ringnumber: 0
        mcastport: 5405
        ttl: 1
    }

    interface {
        ringnumber: 1
        mcastport: 5407
        ttl: 1
    }

    rrp_mode: passive
    transport: udpu

    ...

}

    ...

nodelist {
    node {
            ring0_addr: 192.168.1.11
            ring1_addr: 192.168.2.11
            nodeid: 1
    }

    node {
            ring0_addr: 192.168.1.12
            ring1_addr: 192.168.2.12
            nodeid: 2
    }
}
    ...
```

9.1.3.2 Adapting the SBD configuration

You can skip this section if you do not have any SBD devices, but be sure to implement another supported fencing mechanism.

See the manual pages sbd(8) and stonith_sbd(7) for details.

TABLE 3: SBD OPTIONS IN FILE /ETC/SYSCONFIG/SBD

Parameter	Description
SBD_WATCHDOG_DEV	Define the watchdog device. It is mandatory to use a watchdog. SBD does not work reliable without watchdog. Refer to the SLES manual and SUSE TID 7016880 for setting up a watchdog.
SBD_WATCHDOG_TIMEOUT	This parameter is used with diskless SBD. It defines the timeout, in seconds, the watchdog will wait before panicking the node if noone tickles it. If you set CIB parameter stonith-watchdog-timeout to a negative value, Pacemaker will automatically calculate this timeout and set it to twice the value of SBD_WATCHDOG_TIMEOUT starting with SUSE Linux Enterprise High Availability 15.
SBD_STARTMODE	Start mode. If set to `clean`, sbd will only start if the node was previously shut down cleanly or if the slot is empty.
SBD_PACEMAKER	Check Pacemaker quorum and node health.

In the following, replace /dev/disk/by-id/SBDA and /dev/disk/by-id/SBDB by your real sbd device names. As an example, the `SBD_WATCHDOG_TIMEOUT` is set to 20s to be less aggressive than the formerly used 5s.

```
# egrep -v "(^#|^$)" /etc/sysconfig/sbd
SBD_PACEMAKER=yes
SBD_STARTMODE="clean"
SBD_WATCHDOG_DEV="/dev/watchdog"
SBD_WATCHDOG_TIMEOUT="20"
SBD_TIMEOUT_ACTION="flush,reboot"
SBD_MOVE_TO_ROOT_CGROUP="auto"
SBD_OPTS=""
SBD_DEVICE="/dev/disk/by-id/SBDA;/dev/disk/by-id/SBDB"
```

Important

Also read the SUSE product documentation about calculation of timeouts for more details: https://documentation.suse.com/sle-ha/15-SP1/single-html/SLE-HA-guide/#sec-ha-storage-protect-watchdog-timings

9.1.3.3 Verifying the SBD device

You can skip this section if you do not have any SBD devices, but make sure to implement a supported fencing mechanism.

It is a good practice to check if the SBD device can be accessed from both nodes and does contain valid records. Check this for all devices configured in */etc/sysconfig/sbd*. You can do so, for example, by calling `cs_show_sbd_devices`.

```
suse01:~ # sbd -d /dev/disk/by-id/SBDA -d /dev/disk/by-id/SBDB dump
==Dumping header on disk /dev/disk/by-id/SBDA
Header version     : 2.1
UUID               : 0f4ea13e-fab8-4147-b9b2-3cdcfff07f86
Number of slots    : 255
Sector size        : 512
Timeout (watchdog) : 20
Timeout (allocate) : 2
Timeout (loop)     : 1
Timeout (msgwait)  : 40
==Header on disk /dev/disk/by-id/SBDA is dumped
==Dumping header on disk /dev/disk/by-id/SBDB
Header version     : 2.1
UUID               : 23c423df-675d-4937-a48c-5eb869fe0bb7
Number of slots    : 255
Sector size        : 512
Timeout (watchdog) : 20
Timeout (allocate) : 2
Timeout (loop)     : 1
Timeout (msgwait)  : 40
==Header on disk /dev/disk/by-id/SBDB is dumped
```

Important

The timeout values in our example are only start values. It is a requirement that they are tuned to your environment. Refer to the TIDs 7011346 and 7023689 for more information.

To check the current SBD entries for the various cluster nodes, you can use `sbd list`. If all entries are `clear`, no fencing task is marked in the SBD device.

```
suse01:~ # sbd -d /dev/disk/by-id/SBDA -d /dev/disk/by-id/SBDB list
0     suse01      clear
0     suse01      clear
```

For more information on SBD configuration parameters, consult the respective sections of the SUSE Linux Enterprise High Availability Administration Guide and the TIDs 7016880 and 7008216.

Now it is time to restart the cluster at the first node again (`crm cluster start`).

9.1.4 Configuring the cluster on the second node

The second node of the two nodes cluster can be integrated by starting the command `ha-cluster-join`. This command asks for the IP address or name of the first cluster node. With this command, all needed configuration files are copied over. As a result, the cluster is started on both nodes.

```
# ha-cluster-join -c <host1>
```

Press *RETURN* to acknowledge the IP address.

9.1.5 Checking the cluster for the first time

Now it is time to check and optionally start the cluster for the first time on both nodes.

```
suse01:~ # systemctl status pacemaker
suse01:~ # systemctl status sbd
suse02:~ # systemctl status pacemaker
suse02:~ # systemctl status sbd
suse01:~ # crm cluster start
suse02:~ # crm cluster start
```

Check the cluster status with crm_mon. We use the option "-r" to also see resources, which are configured but stopped.

```
# crm_mon -r1
```

The command will show the "empty" cluster and will print something similar to the screen output below. The most interesting pieces of information for now are that there are two nodes in the status "online", the message "partition with quorum", and a running SBD resource.

```
Cluster Summary:
  * Stack: corosync
  * Current DC: suse01 (version
 2.0.5+20201202.ba59be712-150300.4.16.1-2.0.5+20201202.ba59be712) - partition with quorum
  * Last updated: Thu Jun 10 08:32:58 2022
  * Last change:  Thu Jun 10 08:29:41 2022 by hacluster via crmd on suse01
  * 2 nodes configured
  * 1 resource instance configured

Node List:
  * Online: [ suse01 suse02  ]

Full List of Resources:
  * stonith-sbd (stonith:external/sbd):  Started suse01
```

9.2 Configuring cluster properties and resources

This section describes how to configure constraints, resources, bootstrap, and STONITH, using the `crm configure` shell command as described in part II *Configuration and Administration* of the SUSE Linux Enterprise High Availability Administration Guide.

Use the command `crm` to add the objects to the cluster information base (CIB). Copy the following examples to a local file, edit the file and then load the configuration to the CIB:

```
suse01:~ # vi crm-fileXX
suse01:~ # crm configure load update crm-fileXX
```

9.2.1 Cluster bootstrap and more

The first example defines the cluster bootstrap options, the resource and operation defaults. The stonith-timeout should be greater than 1.2 times the SBD on-disk msgwait timeout. The priority-fencing-delay should be at least 2 times the SBD CIB pcmk_delay_max.

```
suse01:~ # vi crm-bs.txt
# enter the following to crm-bs.txt
property cib-bootstrap-options: \
    stonith-enabled="true" \
    stonith-action="reboot" \
    stonith-timeout="150" \
    priority-fencing-delay="30"
rsc_defaults rsc-options: \
    resource-stickiness="1000" \
```

```
    migration-threshold="5000"
op_defaults op-options: \
    timeout="600" \
    record-pending=true
```

Now add the configuration to the cluster.

```
suse01:~ # crm configure load update crm-bs.txt
```

9.2.2 STONITH device

Skip this section if you are using diskless SBD.

The next configuration part defines an SBD disk STONITH resource.

```
# vi crm-sbd.txt
# enter the following to crm-sbd.txt
primitive stonith-sbd stonith:external/sbd \
    params pcmk_delay_max="15"
```

Again we add the configuration to the cluster.

```
suse01:~ # crm configure load update crm-sbd.txt
```

For fencing with IPMI/ILO see section *Section 9.2.3, "Using IPMI as fencing mechanism"*.

9.2.3 Using IPMI as fencing mechanism

For details about IPMI/ILO fencing see our cluster product documentation (https://documentation.suse.com/sle-ha/15-SP6/single-html/SLE-HA-administration/). An example for an IPMI STONITH resource can be found in section *Section 12.4.2, "Example for the IPMI STONITH method"* of this document.

To use IPMI, the remote management boards must be compatible with the IPMI standard.

For the IPMI-based fencing, configure a primitive per-cluster node. Each resource is responsible to fence exactly one cluster node. Adapt the IP addresses and login user / password of the remote management boards to the STONITH resource agent. We recommend to create a special STONITH user instead of providing root access to the management board. Location rules must guarantee that a host should never run its own STONITH resource.

9.2.4 Using other fencing mechanisms

This section is only relevant if the recommended disk-based or diskless SBD fencing is not used.

We recommend to use SBD (best practice) or IPMI (second choice) as STONITH mechanism. The SUSE Linux Enterprise High Availability product also supports additional fencing mechanism not covered here.

For further information about fencing, read the Administration Guide for SUSE Linux Enterprise High Availability at https://documentation.suse.com/sle-ha/15-SP6/single-html/SLE-HA-administration/. For public cloud environments, refer to your cloud provider's documentation on supported fencing mechanisms.

9.2.5 SAPHanaTopology

This step is to define the resources needed, to analyze the SAP HANA topology for the replicated pair. Prepare the changes in a text file, for example *crm-saphanatop.txt*, and load it with the command:

```
crm configure load update crm-saphanatop.txt
```

```
# vi crm-saphanatop.txt
# enter the following to crm-saphanatop.txt
primitive rsc_SAPHanaTop_HA1_HDB10 ocf:suse:SAPHanaTopology \
    op start interval=0 timeout=600 \
    op stop interval=0 timeout=300 \
    op monitor interval=50 timeout=600 \
    params SID=HA1 InstanceNumber=10
clone cln_SAPHanaTop_HA1_HDB10 rsc_SAPHanaTop_HA1_HDB10 \
    meta clone-node-max=1 interleave=true
```

Additional information about all parameters can be found in manual page ocf_suse_SAPHanaTopology(7).

Again, add the configuration to the cluster.

```
suse01:~ # crm configure load update crm-saphanatop.txt
```

The most important parameters here are SID and InstanceNumber, which are quite self explaining in the SAP context. Beside these parameters, typical tuneables are the timeout values or the operations (start, monitor, stop).

9.2.6 SAPHanaFilesystem

This step is to define the resources to monitor the filesystem used by HANA, e.g. /hana/shared/<SID>. The RA just monitors the filesystem, but neither does mount nor umount it. Mounting and umounting is done by the OS thru /etc/fstab. Prepare the changes in a text file, for example *crm-saphanafil.txt*, and load it with the command:

```
crm configure load update crm-saphanafil.txt
```

```
# vi crm-saphanafil.txt
# enter the following to crm-saphanafil.txt
primitive rsc_SAPHanaFil_HA1_HDB10 ocf:suse:SAPHanaFilesystem \
    op start interval=0 timeout=10 \
    op stop interval=0 timeout=20 \
    op monitor interval=120 timeout=120 \
    params SID=HA1 InstanceNumber=10 ON_FAIL_ACTION="fence"
clone cln_SAPHanaFil_HA1_HDB10 rsc_SAPHanaFil_HA1_HDB10 \
    meta clone-node-max=1 interleave=true
```

Additional information about all parameters can be found in manual page ocf_suse_SAPHanaFilesystem(7).

Again, add the configuration to the cluster.

```
suse01:~ # crm configure load update crm-saphanafil.txt
```

The most important parameters here are SID and InstanceNumber, which are quite self explaining in the SAP context. ON_FAIL_ACTION defines how the RA should react on monitor failures. Beside these parameters, typical tuneables are the timeout values or the operations (start, monitor, stop).

9.2.7 SAPHanaController

This step is to define the resource needed, to control the replicated SAP HANA pair. Edit the changes in a text file, for example *crm-saphanacon.txt*, and load it with the following command:

```
crm configure load update crm-saphanacon.txt
```

TABLE 4: TYPICAL RESOURCE AGENT PARAMETER SETTINGS FOR DIFFERENT SCENARIOS

Parameter	Performance Optimized	Cost Optimized	Multi-Tier
PREFER_SITE_TAKEOVER	true	false	false / true

Parameter	Performance Optimized	Cost Optimized	Multi-Tier
AUTOMATED_REGISTER	false / true	false / true	false
DUPLICATE_PRIMARY_TIMEOUT	7200	7200	7200

TABLE 5: DESCRIPTION OF IMPORTANT RESOURCE AGENT PARAMETERS

Parameter	Description
PREFER_SITE_TAKEOVER	Defines whether RA should prefer to take over to the secondary instance instead of restarting the failed primary locally.
AUTOMATED_REGISTER	Defines whether a former primary should be automatically registered to be secondary of the new primary. With this parameter you can adapt the level of system replication automation. If set to `false`, the former primary must be manually registered. The cluster will not start this SAP HANA RDBMS until it is registered, to avoid double primary up situations.
DUPLICATE_PRIMARY_TIMEOUT	Time difference needed between two primary time stamps if a dual-primary situation occurs. If the time difference is less than the time gap, the cluster holds one or both instances in a "WAITING" status. This is to give an administrator the chance to react on a failover. If the complete node of the former primary crashed, the former primary will be registered after the time difference is passed. If "only" the SAP HANA RDBMS has crashed, the former primary will be registered immediately. After this registration to the new primary, all data will be overwritten by the system replication.

Additional information about all parameters of the SAPHanaController RA can be found in manual page ocf_suse_SAPHanaController(7).

```
# vi crm-saphanacon.txt
# enter the following to crm-saphanacon.txt
primitive rsc_SAPHanaCon_HA1_HDB10 ocf:suse:SAPHanaController \
    op start interval=0 timeout=3600 \
    op stop interval=0 timeout=3600 \
    op promote interval=0 timeout=900 \
    op demote interval=0 timeout=320 \
    op monitor interval=60 role=Promoted timeout=700 \
    op monitor interval=61 role=Unpromoted timeout=700 \
    params SID=HA1 InstanceNumber=10 PREFER_SITE_TAKEOVER=true \
        DUPLICATE_PRIMARY_TIMEOUT=7200 AUTOMATED_REGISTER=false \
    meta priority=100
clone mst_SAPHanaCon_HA1_HDB10 rsc_SAPHanaCon_HA1_HDB10 \
    meta clone-node-max=1 promotable=true interleave=true maintenance=true
```

Now add the configuration to the cluster.

```
suse01:~ # crm configure load update crm-saphanacon.txt
```

The most important parameters here are again SID and InstanceNumber. Beside these parameters the timeout values for the operations (start, promote, monitors, stop) are typical tuneables.

9.2.8 Adding a virtual IP address for the primary site

The last resource to be added is covering the virtual IP address. For details, see manual page ocf_heartbeat_IPaddr2(7).

```
# vi crm-vip.txt
# enter the following to crm-vip.txt

primitive rsc_ip_HA1_HDB10 ocf:heartbeat:IPaddr2 \
    op monitor interval=10 timeout=20 \
    params ip=192.168.1.20
```

Load the file to the cluster.

```
suse01:~ # crm configure load update crm-vip.txt
```

In most on-premise installations, only the parameter *ip* needs to be set to the virtual IP address to be presented to the client systems. Public cloud environments often need specific settings.

9.2.9 Constraints for SAPHanaSR-angi

Two constraints are organizing the correct placement of the virtual IP address for the client database access and the start order between the resource agents SAPHanaController and SAPHanaTopology.

```
# vi crm-cs.txt
# enter the following to crm-cs.txt
colocation col_saphana_ip_HA1_HDB10 2000: rsc_ip_HA1_HDB10:Started \
    mst_SAPHanaCon_HA1_HDB10:Promoted
order ord_saphana_HA1_HDB10 Optional: cln_SAPHanaTop_HA1_HDB10 \
    mst_SAPHanaCon_HA1_HDB10
```

Load the file to the cluster.

```
suse01:~ # crm configure load update crm-cs.txt
```

9.2.10 Activating multi-state resource for cluster operation

The multi state resource was added to the cluster with in maintenance mode. To get the resource operated by the cluster the maintenance must be ended by the command:

```
# crm resource refresh mst_SAPHanaCon_HA1_HDB10
# cs_wait_for_idle -s 5
# crm resource maintenance mst_SAPHanaCon_HA1_HDB10 off
```

The command `cs_wait_for_idle` is part of the package `ClusterTools2`. For more details, see manual pages cs_wait_for_idle(8), crm(8), SAPHanaSR_maintenance_examples(7).

9.2.11 Active/active read-enabled scenario

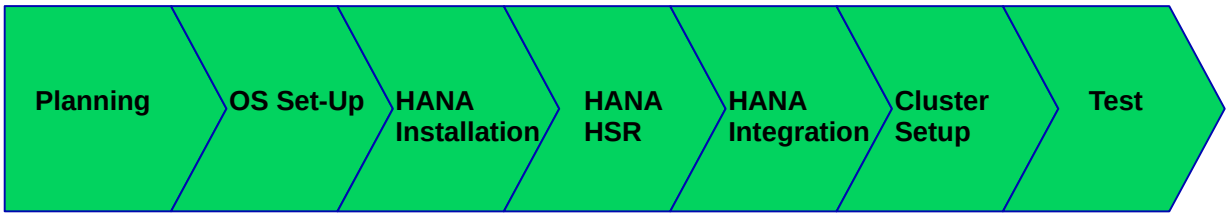
This step is optional. If you have an active/active SAP HANA system replication with a read-enabled secondary, it is possible to integrate the needed second virtual IP address into the cluster. This is been done by adding a second virtual IP address resource and a location constraint binding the address to the secondary site.

```
# vi crm-re.txt
# enter the following to crm-re.txt

primitive rsc_ip_HA1_HDB10_readenabled ocf:heartbeat:IPaddr2 \
    op monitor interval=10 timeout=20 \
    params ip=192.168.1.21
```

```
colocation col_saphana_ip_HA1_HDB10_readenabled 2000: \
    rsc_ip_HA1_HDB10_readenabled:Started mst_SAPHanaCon_HA1_HDB10:Unpromoted
```

10 Testing the cluster



The lists of tests will be further enhanced in one of the next updates of this document.

As with any cluster testing is crucial. Make sure that all test cases derived from customer expectations are implemented and passed fully. Otherwise the project is likely to fail in production.

The test prerequisite, if not described differently, is always that both nodes are booted, normal members of the cluster and the HANA RDBMS is running. The system replication is in sync (SOK).

10.1 Test cases for semi-automation

In the following test descriptions we assume `PREFER_SITE_TAKEOVER="true"` and `AUTOMATED_REGISTER="false"`.

Note

The following tests are designed to be run in sequence and depend on the exit state of the preceding tests.

10.1.1 Test: Stop primary database on site A (node 1)

EXAMPLE 19: TEST STOP_PRIMARY_SITE_A

COMPONENT:

- Primary Database

DESCRIPTION:

- The primary SAP HANA database is stopped during normal cluster operation.

TEST PROCEDURE:

1. Stop the primary SAP HANA database gracefully as *<sid>adm*.

```
suse01:~> HDB stop
```

RECOVERY PROCEDURE:

1. Manually register the old primary (on node 1) with the new primary after takeover (on node 2) as *<sid>adm*.

```
suse01:~> hdbnsutil -sr_register --remoteHost=suse02 --remoteInstance=10 \
          --replicationMode=sync --operationMode=logreplay \
          --name=WDF
```

2. Restart the SAP HANA database (now secondary) on node 1 as root.

```
# crm resource refresh rsc_SAPHanaCon_HA1_HDB10 suse01
```

EXPECTED:

1. The cluster detects the stopped primary SAP HANA database (on node 1) and marks the resource failed.
2. The cluster promotes the secondary SAP HANA database (on node 2) to take over as primary.
3. The cluster migrates the IP address to the new primary (on node 2).
4. After some time the cluster shows the sync_state of the stopped primary (on node 1) as SFAIL.
5. Because AUTOMATED_REGISTER = "false" the cluster does not restart the failed SAP HANA database or register it against the new primary.
6. After the manual register and resource refresh the system replication pair is marked as in sync (SOK).
7. The cluster "failed actions" are cleaned up after following the recovery procedure.

10.1.2 Test: Stop primary database on site B (node 2)

EXAMPLE 20: TEST STOP_PRIMARY_DB_SITE_B

Component:

Primary Database

Description:

The primary SAP HANA database is stopped during normal cluster operation.

TEST PROCEDURE:

1. Stop the database gracefully as <sid>adm.

```
suse02:~> HDB stop
```

RECOVERY PROCEDURE:

1. Manually register the old primary (on node 2) with the new primary after takeover (on node 1) as *<sid>adm*.

```
suse02:~> hdbnsutil -sr_register --remoteHost=suse01 --remoteInstance=10 \
          --replicationMode=sync --operationMode=logreplay \
          --name=ROT
```

2. Restart the SAP HANA database (now secondary) on node 2 as root.

```
# crm resource refresh rsc_SAPHanaCon_HA1_HDB10 suse02
```

EXPECTED:

1. The cluster detects the stopped primary SAP HANA database (on node 2) and marks the resource failed.
2. The cluster promotes the secondary SAP HANA database (on node 1) to take over as primary.
3. The cluster migrates the IP address to the new primary (on node 1).
4. After some time the cluster shows the sync_state of the stopped primary (on node 2) as SFAIL.
5. Because AUTOMATED_REGISTER = "false" the cluster does not restart the failed SAP HANA database or register it against the new primary.
6. After the manual register and resource refresh the system replication pair is marked as in sync (SOK).
7. The cluster "failed actions" are cleaned up after following the recovery procedure.

10.1.3 Test: Crash primary database on site A (node 1)

EXAMPLE 21: TEST CRASH_PRIMARY_DB_SITE_A

Component:

Primary Database

Description:

Simulate a complete break-down of the primary database system.

TEST PROCEDURE:

1. Kill the primary database system using signals as *<sid>adm*.

```
suse01:~> HDB kill-9
```

RECOVERY PROCEDURE:

1. Manually register the old primary (on node 1) with the new primary after takeover (on node 2) as *<sid>adm*.

```
suse01:~> hdbnsutil -sr_register --remoteHost=suse02 --remoteInstance=10 \
          --replicationMode=sync  --operationMode=logreplay \
          --name=WDF
```

2. Restart the SAP HANA database (now secondary) on node 1 as root.

```
# crm resource refresh rsc_SAPHanaCon_HA1_HDB10 suse01
```

EXPECTED:

1. The cluster detects the stopped primary SAP HANA database (on node 1) and marks the resource failed.
2. The cluster promotes the secondary SAP HANA database (on node 2) to take over as primary.
3. The cluster migrates the IP address to the new primary (on node 2).
4. After some time the cluster shows the sync_state of the stopped primary (on node 1) as SFAIL.
5. Because AUTOMATED_REGISTER = "false" the cluster does not restart the failed SAP HANA database or register it against the new primary.

6. After the manual register and resource refresh the system replication pair is marked as in sync (SOK).
7. The cluster "failed actions" are cleaned up after following the recovery procedure.

10.1.4 Test: Crash primary database on site B (node 2)

EXAMPLE 22: TEST CRASH_PRIMARY_DB_SITE_B

Component:

Primary Database

Description:

Simulate a complete break-down of the primary database system.

TEST PROCEDURE:

1. Kill the primary database system using signals as *<sid>adm*.

```
suse02:~> HDB kill-9
```

RECOVERY PROCEDURE:

1. Manually register the old primary (on node 2) with the new primary after takeover (on node 1) as *<sid>adm*.

```
suse02:~> hdbnsutil -sr_register --remoteHost=suse01 --remoteInstance=10 \
          --replicationMode=sync  --operationMode=logreplay \
          --name=ROT
```

2. Restart the SAP HANA database (now secondary) on node 2 as root.

```
# crm resource refresh rsc_SAPHanaCon_HA1_HDB10 suse02
```

EXPECTED:

1. The cluster detects the stopped primary SAP HANA database (on node 2) and marks the resource failed.
2. The cluster promotes the secondary SAP HANA database (on node 1) to take over as primary.
3. The cluster migrates the IP address to the new primary (on node 1).

4. After some time the cluster shows the sync_state of the stopped primary (on node 2) as SFAIL.
5. Because AUTOMATED_REGISTER = "false" the cluster does not restart the failed SAP HANA database or register it against the new primary.
6. After the manual register and resource refresh the system replication pair is marked as in sync (SOK).
7. The cluster "failed actions" are cleaned up after following the recovery procedure.

10.1.5 Test: Crash primary node on site A (node 1)

EXAMPLE 23: TEST CRASH_PRIMARY_NODE_SITE_A

Component:

Cluster node of primary site

Description:

Simulate a crash of the primary site node running the primary SAP HANA database.

TEST PROCEDURE:

1. Crash the primary node by sending a 'fast-reboot' system request.

```
suse01:~ # sync; echo b > /proc/sysrq-trigger
```

RECOVERY PROCEDURE:

1. If SBD fencing is used, pacemaker will not automatically restart after being fenced. In this case clear the fencing flag on all SBD devices and subsequently start pacemaker.

```
suse01:~ # sbd -d /dev/disk/by-id/SBDA message suse01 clear
suse01:~ # sbd -d /dev/disk/by-id/SBDB message suse01 clear
...
```

2. Start the cluster framework

```
suse01:~ # crm cluster start
```

3. Manually register the old primary (on node 1) with the new primary after takeover (on node 2) as *<sid>adm*.

```
suse01:~> hdbnsutil -sr_register --remoteHost=suse02 --remoteInstance=10 \
          --replicationMode=sync  --operationMode=logreplay \
          --name=WDF
```

4. Restart the SAP HANA database (now secondary) on node 1 as root.

```
# crm resource refresh rsc_SAPHanaCon_HA1_HDB10 suse01
```

EXPECTED:

1. The cluster detects the failed node (node 1) and declares it UNCLEAN and sets the secondary node (node 2) to status "partition with quorum".
2. The cluster fences the failed node (node 1).
3. The cluster declares the failed node (node 1) OFFLINE.
4. The cluster promotes the secondary SAP HANA database (on node 2) to take over as primary.
5. The cluster migrates the IP address to the new primary (on node 2).
6. After some time the cluster shows the sync_state of the stopped primary (on node 2) as SFAIL.
7. If SBD fencing is used, then the manual recovery procedure will be used to clear the fencing flag and restart pacemaker on the node.
8. Because AUTOMATED_REGISTER = "false" the cluster does not restart the failed SAP HANA database or register it against the new primary.
9. After the manual register and resource refresh the system replication pair is marked as in sync (SOK).
10. The cluster "failed actions" are cleaned up after following the recovery procedure.

10.1.6 Test: Crash primary node on site B (node 2)

EXAMPLE 24: TEST CRASH_PRIMARY_NODE_SITE_B

Component:

Cluster node of secondary site

Description:

Simulate a crash of the secondary site node running the primary SAP HANA database.

TEST PROCEDURE:

1. Crash the secondary node by sending a 'fast-reboot' system request.

```
suse02:~ # sync; echo b > /proc/sysrq-trigger
```

RECOVERY PROCEDURE:

1. If SBD fencing is used, pacemaker will not automatically restart after being fenced. In this case clear the fencing flag on all SBD devices and subsequently start pacemaker.

```
suse02:~ # sbd -d /dev/disk/by-id/SBDA message suse02 clear
suse02:~ # sbd -d /dev/disk/by-id/SBDB message suse02 clear
...
```

2. Start the cluster Framework

```
suse02:~ # crm cluster start
```

3. Manually register the old primary (on node 2) with the new primary after takeover (on node 1) as *<sid>adm*.

```
suse02:~> hdbnsutil -sr_register --remoteHost=suse01 --remoteInstance=10 \
          --replicationMode=sync  --operationMode=logreplay \
          --name=ROT
```

4. Restart the SAP HANA database (now secondary) on node 2 as root.

```
# crm resource refresh rsc_SAPHanaCon_HA1_HDB10 suse02
```

EXPECTED:

1. The cluster detects the failed secondary node (node 2) and declares it UNCLEAN and sets the primary node (node 1) to status "partition with quorum".
2. The cluster fences the failed secondary node (node 2).
3. The cluster declares the failed secondary node (node 2) OFFLINE.
4. The cluster promotes the secondary SAP HANA database (on node 1) to take over as primary.
5. The cluster migrates the IP address to the new primary (on node 1).

6. After some time the cluster shows the sync_state of the stopped secondary (on node 2) as SFAIL.
7. If SBD fencing is used, then the manual recovery procedure will be used to clear the fencing flag and restart pacemaker on the node.
8. Because AUTOMATED_REGISTER = "false" the cluster does not restart the failed SAP HANA database or register it against the new primary.
9. After the manual register and resource refresh the system replication pair is marked as in sync (SOK).
10. The cluster "failed actions" are cleaned up after following the recovery procedure.

10.1.7 Test: Stop the secondary database on site B (node 2)

EXAMPLE 25: TEST STOP_SECONDARY_DB_SITE_B

Component:

Secondary SAP HANA database

Description:

The secondary SAP HANA database is stopped during normal cluster operation.

TEST PROCEDURE:

1. Stop the secondary SAP HANA database gracefully as *<sid>adm*.

```
suse02:~> HDB stop
```

RECOVERY PROCEDURE:

1. Refresh the failed resource status of the secondary SAP HANA database (on node 2) as root.

```
# crm resource refresh rsc_SAPHanaCon_HA1_HDB10 suse02
```

EXPECTED:

1. The cluster detects the stopped secondary database (on node 2) and marks the resource failed.
2. The cluster detects the broken system replication and marks it as failed (SFAIL).

3. The cluster restarts the secondary SAP HANA database on the same node (node 2).
4. The cluster detects that the system replication is in sync again and marks it as ok (SOK).
5. The cluster "failed actions" are cleaned up after following the recovery procedure.

10.1.8 Test: Crash the secondary database on site B (node 2)

EXAMPLE 26: TEST CRASH_SECONDARY_DB_SITE_B

Component:

Secondary SAP HANA database

Description:

Simulate a complete break-down of the secondary database system.

TEST PROCEDURE:

1. Kill the secondary database system using signals as *<sid>adm*.

```
suse02:~> HDB kill-9
```

RECOVERY PROCEDURE:

1. Clean up the failed resource status of the secondary SAP HANA database (on node 2) as root.

```
# crm resource refresh rsc_SAPHanaCon_HA1_HDB10 suse02
```

EXPECTED:

1. The cluster detects the stopped secondary database (on node 2) and marks the resource failed.
2. The cluster detects the broken system replication and marks it as failed (SFAIL).
3. The cluster restarts the secondary SAP HANA database on the same node (node 2).
4. The cluster detects that the system replication is in sync again and marks it as ok (SOK).
5. The cluster "failed actions" are cleaned up after following the recovery procedure.

10.1.9 Test: Crash the secondary node on site B (node2)

EXAMPLE 27: TEST CRASH_SECONDARY_NODE_SITE_B

Component:

Cluster node of secondary site

Description:

Simulate a crash of the secondary site node running the secondary SAP HANA database.

TEST PROCEDURE:

1. Crash the secondary node by sending a 'fast-reboot' system request.

```
suse02:~ # sync; echo b > /proc/sysrq-trigger
```

RECOVERY PROCEDURE:

1. If SBD fencing is used, pacemaker will not automatically restart after being fenced. In this case clear the fencing flag on **all** SBD devices and subsequently start pacemaker.

```
suse02:~ # sbd -d /dev/disk/by-id/SBDA message suse02 clear
suse02:~ # sbd -d /dev/disk/by-id/SBDB message suse02 clear
...
```

2. Start the cluster framework.

```
suse02:~ # crm cluster start
```

EXPECTED:

1. The cluster detects the failed secondary node (node 2) and declares it UNCLEAN and sets the primary node (node 1) to status "partition with quorum".
2. The cluster fences the failed secondary node (node 2).
3. The cluster declares the failed secondary node (node 2) OFFLINE.
4. After some time the cluster shows the sync_state of the stopped secondary (on node 2) as SFAIL.
5. If SBD fencing is used, then the manual recovery procedure will be used to clear the fencing flag and restart pacemaker on the node.

6. When the fenced node (node 2) rejoins the cluster the former secondary SAP HANA database is started automatically.
7. The cluster detects that the system replication is in sync again and marks it as ok (SOK).

10.1.10 Test: Failure of replication LAN

EXAMPLE 28: TEST FAIL_NETWORK_SR

Component:

Replication LAN

Description:

Loss of replication LAN connectivity between the primary and secondary node.

TEST PROCEDURE:

1. Break the connection between the cluster nodes on the replication LAN.

RECOVERY PROCEDURE:

1. Re-establish the connection between the cluster nodes on the replication LAN.

EXPECTED:

1. After some time the cluster shows the sync_state of the secondary (on node 2) as SFAIL.
2. The primary SAP HANA database (node 1) "HDBSettings.sh systemReplicationStatus.py" shows "CONNECTION TIMEOUT" and the secondary SAP HANA database (node 2) is not able to reach the primary database (node 1).
3. The primary SAP HANA database continues to operate as “normal”, but no system replication takes place and is therefore no longer a valid take over destination.
4. When the LAN connection is re-established, HDB automatically detects connectivity between the SAP HANA databases and restarts the system replication process
5. The cluster detects that the system replication is in sync again and marks it as ok (SOK).

10.2 Test cases for full automation

In the following test descriptions we assume **PREFER_SITE_TAKEOVER** = **"true"** and **AUTOMATED_REGISTER** = **"true".**

Note

The following tests are designed to be run in sequence and depend on the exit state of the preceding tests.

10.2.1 Test: Stop the primary database on site A

EXAMPLE 29: TEST STOP_PRIMARY_DB_SITE_A

COMPONENT:

- Primary Database

DESCRIPTION:

- The primary SAP HANA database is stopped during normal cluster operation.

TEST PROCEDURE:

- Stop the primary SAP HANA database gracefully as *<sid>adm*.

```
suse01:~> HDB stop
```

RECOVERY PROCEDURE:

1. Not needed, everything is automated
2. Refresh the cluster resources on node 1 as root.

```
# crm resource refresh rsc_SAPHanaCon_HA1_HDB10 suse01
```

EXPECTED:

1. The cluster detects the stopped primary SAP HANA database (on node 1) and marks the resource failed.
2. The cluster promotes the secondary SAP HANA database (on node 2) to take over as primary.
3. The cluster migrates the IP address to the new primary (on node 2).

4. After some time the cluster shows the sync_state of the stopped primary (on node 1) as SFAIL.
5. Because AUTOMATED_REGISTER = "true" the cluster does restart the failed SAP HANA database and register it against the new primary.
6. After the automated register and resource refresh the system replication pair is marked as in sync (SOK).
7. The cluster "failed actions" are cleaned up after following the recovery procedure.

10.2.2 Test: Crash the primary node on site B (node 2)

EXAMPLE 30: TEST CRASH_PRIMARY_NODE_SITE_B

COMPONENT:

- Cluster node of site B

DESCRIPTION:

- Simulate a crash of the site B node running the primary SAP HANA database.

TEST PROCEDURE:

- Crash the secondary node by sending a 'fast-reboot' system request.

```
suse02:~ # sync; echo b > /proc/sysrq-trigger
```

RECOVERY PROCEDURE:

- If SBD fencing is used, pacemaker will not automatically restart after being fenced. In this case clear the fencing flag on **all** SBD devices and subsequently start pacemaker.

```
suse02:~ # sbd -d /dev/disk/by-id/SBDA message suse02 clear
suse02:~ # sbd -d /dev/disk/by-id/SBDB message suse02 clear
...
```

- Start the cluster framework.

```
suse02:~ # crm cluster start
```

- Refresh the cluster resources on node 2 as root.

```
# crm resource refresh rsc_SAPHanaCon_HA1_HDB10 suse02
```

EXPECTED:

1. The cluster detects the failed primary node (node 2) and declares it UNCLEAN and sets the primary node (node 2) to status "partition with quorum".
2. The cluster fences the failed primary node (node 2).
3. The cluster declares the failed primary node (node 2) OFFLINE.
4. The cluster promotes the secondary SAP HANA database (on node 1) to take over as primary.
5. The cluster migrates the IP address to the new primary (on node 1).
6. After some time the cluster shows the sync_state of the stopped secondary (on node 2) as SFAIL.
7. If SBD fencing is used, then the manual recovery procedure will be used to clear the fencing flag and restart pacemaker on the node.
8. When the fenced node (node 2) rejoins the cluster the former primary became a secondary.
9. Because AUTOMATED_REGISTER = "true" the cluster does restart the failed SAP HANA database and register it against the new primary.
10. The cluster detects that the system replication is in sync again and marks it as ok (SOK).

11 Administration

11.1 Dos and don’ts

In your project, you should:

- Define STONITH before adding other resources to the cluster.
- Do intensive testing.
- Tune the timeouts of operations of SAPHana and SAPHanaTopology.
- Start with the parameter values PREFER_SITE_TAKEOVER = ”true”, AUTOMATED_REGISTER = ”false” and DUPLICATE_PRIMARY_TIMEOUT = ”7200”.

- Always wait for pending cluster actions to finish before doing something.
- Set up a test cluster for testing configuration changes and administrative procedure before applying them on the production cluster.

In your project, avoid:

- Rapidly changing/changing back a cluster configuration, such as setting nodes to standby and online again or stopping/starting the multi-state resource.
- Creating a cluster without proper time synchronization or unstable name resolutions for hosts, users and groups.
- Using site names other than the ones already known by the cluster when manually re-registering a site.
- Adding location rules for the clone, multi-state or IP resource. Only location rules mentioned in this setup guide are allowed. For public cloud refer to the cloud specific documentation.
- Using SAP tools for attempting start/stop/takeover actions on a database while the cluster is in charge of managing that database. Same for unregistering/disabling system replication.

Important

As "migrating" or "moving" resources in crm-shell, HAWK or other tools would add client-prefer location rules, support is limited to maintenance procedures described in this document. See *Section 10, "Testing the cluster"* and *Section 11.3, "Maintenance"* (page 70) for proven procedures.

11.2 Monitoring and tools

You can use the High Availability Web Console (HAWK), SAP HANA Cockpit, SAP HANA Studio and different command line tools for cluster status requests.

11.2.1 HAWK – cluster status and more

You can use a Web browser to check the cluster status.

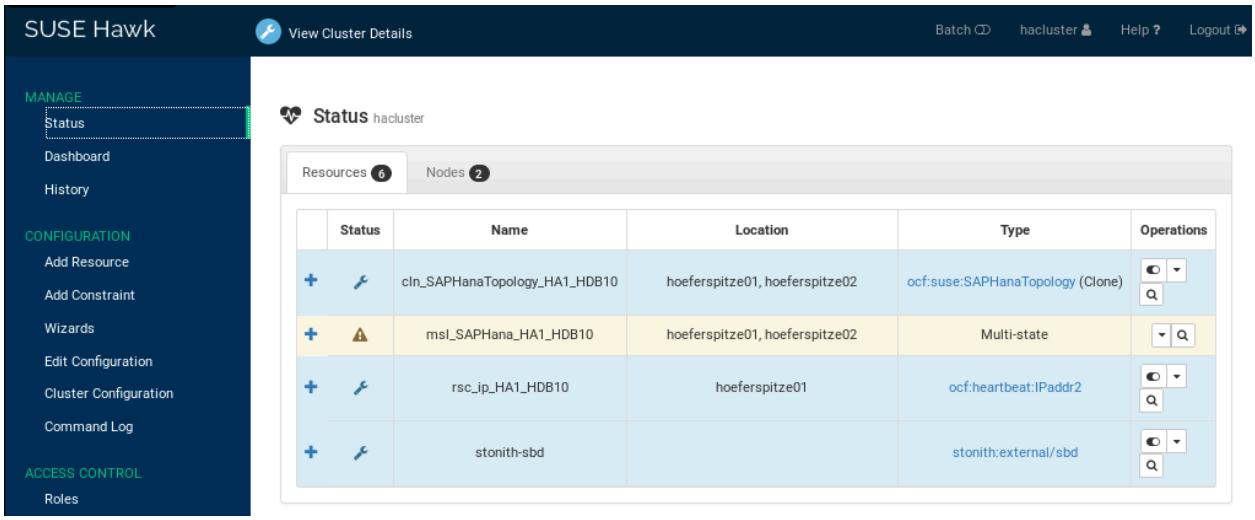


FIGURE 9: CLUSTER STATUS IN HAWK

If you set up the cluster using ha-cluster-init and you have installed all packages as described above, your system will provide a very useful Web interface. You can use this graphical Web interface to get an overview of the complete cluster status, perform administrative tasks or configure resources and cluster bootstrap parameters. Read the product manuals for a complete documentation of this user interface. For the SAP HANA system replication performance optimized scenario the use of HAWK should follow the guidance given in this guide.

11.2.2 SAP HANA Cockpit

Database-specific administration and checks can be done with SAP HANA Cockpit. Before trying start/stop/takeover for the database, make sure the cluster is not in charge of managing the respective resource. See also *Section 11.3, "Maintenance"* (page 70).

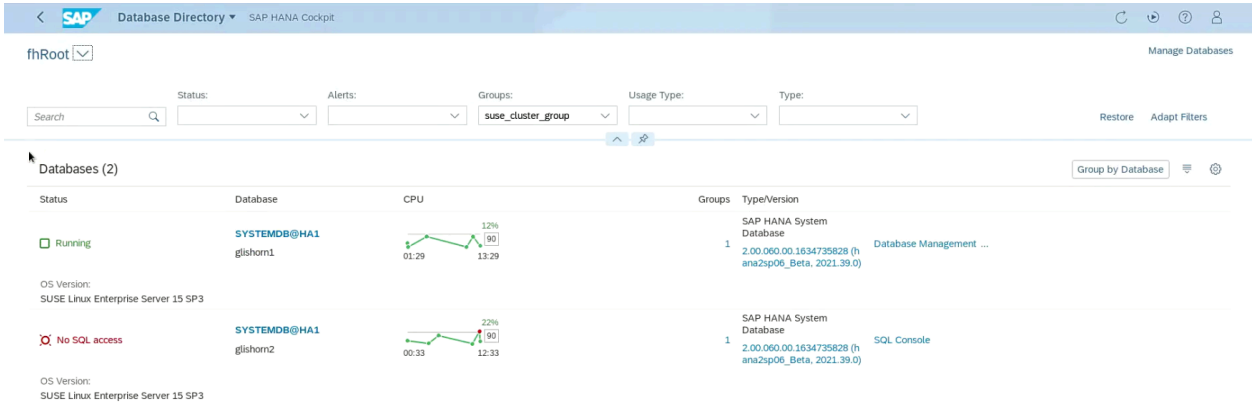


FIGURE 10: SAP HANA COCKPIT – DATABASE DIRECTORY

11.2.3 Cluster command line tools

A simple overview can be obtained by calling `crm_mon`. Using option `-r` shows also stopped but already configured resources. Option `-1` tells `crm_mon` to output the status once instead of periodically.

```
# crm_mon -1r
Stack: corosync
Current DC: suse01 (version 2.0.1+20190417.13d370ca9-3.6.1-2.0.1+20190417.13d370ca9) -
 partition with quorum
Last updated: Thu Feb  6 12:20:03 2024
Last change: Thu Feb  6 12:19:43 2024 by root via crm_attribute on suse01

2 nodes configured
6 resources configured

Online: [ suse01 suse02 ]

Full list of resources:

 stonith-sbd    (stonith:external/sbd): Started suse01
 Clone Set: cln_SAPHanaTop_HA1_HDB10 [rsc_SAPHanaTop_HA1_HDB10]
     Started: [ suse01 suse02 ]
 Clone Set: cln_SAPHanaFil_HA1_HDB10 [rsc_SAPHanaFil_HA1_HDB10]
     Started: [ suse01 suse02 ]
 Clone Set: mst_SAPHanaCon_HA1_HDB10 [rsc_SAPHanaCon_HA1_HDB10] (promotable)
     Masters: [ suse01 ]
     Slaves: [ suse02 ]
 rsc_ip_HA1_HDB10       (ocf::heartbeat:IPaddr2):       Started suse01
```

See the manual page crm_mon(8) for details.

11.2.4 SAPHanaSR command line tools

To show SAPHanaController and SAPHanaTopology resource agent internal values, you can call the program `SAPHanaSR-showAttr`. The internal values, the attribute location and their parameter names may change. The command `SAPHanaSR-showAttr` will always fetch the values from the correct attribute location. Do not use cluster commands like `crm_attribute` to fetch the values directly from the cluster. *SAPHanaSR-showAttr* should not be used for automated system monitoring.

```
suse01:~ # SAPHanaSR-showAttr
Global cib-time                 prim sec sid topology
-----------------------------------------------------
```

```
global Mon May 08 09:10:11 2023 WDF  JWD HA1 ScaleUp

Resource              maintenance
---------------------------------
mst_SAPHana_HA1_HDB10 false

Sites lpt        lss mns    opMode    srHook srMode srPoll srr
-----------------------------------------------------------------
ROT   30         4   suse02 logreplay SOK    sync   SOK    S
WDF   1683529811 4   suse01 logreplay PRIM   sync   PRIM   P

Hosts  clone_state roles                        score site srah
-----------------------------------------------------------------
suse01 PROMOTED    master1:master:worker:master 150   WDF  -
suse02 DEMOTED     master1:master:worker:master 100   ROT  -
```

`SAPHanaSR-showAttr` also supports other output formats such as **script**. The script format is intended to allow running filters. Based on `SAPHanaSR-showAttr` output format script you can define effective queries:

```
suse01:~ # SAPHanaSR-showAttr --format=script | grep "/remoteHost="
Thu Feb  6 12:28:10 2020; Hosts/suse01/remoteHost=suse02
Thu Feb  6 12:28:10 2020; Hosts/suse02/remoteHost=suse01
```

`SAPHanaSR-replay-archive` can help to analyze the SAPHanaSR attribute values from `crm_report` archives. This allows post mortem analysis.

In our example, the administrator killed the primary SAP HANA instance using the command `HDB kill-9`. This happened around 9:10 pm.

```
suse01:~ # hb_report -f 19:00
INFO: suse01# The report is saved in ./hb_report-1-11-11-2019.tar.bz2
INFO: suse01# Report timespan: 11/11/19 19:00:00 - 11/11/19 21:05:33
INFO: suse01# Thank you for taking time to create this report.
suse01:~ # SAPHanaSR-replay-archive --format=script \
    ./hb_report-1-11-11-2019.tar.bz2 | grep "/roles="
Mon Nov 11 20:38:01 2019; Hosts/suse01/roles=master1:master:worker:master
Mon Nov 11 20:38:01 2019; Hosts/suse02/roles=master1:master:worker:master
Mon Nov 11 21:11:37 2019; Hosts/suse01/roles=master1::worker:
Mon Nov 11 21:12:43 2019; Hosts/suse02/roles=master1:master:worker:master
```

In the above example the attributes indicate that at the beginning suse01 was running primary (4:P) and suse02 was running secondary (4:S).

At 21:11 (CET) suddenly the primary on suse01 died - it was falling down to 1:P.

The cluster did jump-in and initiated a takeover. At 21:12 (CET) the former secondary was detected as new running master (changing from 4:S to 4:P).

11.2.5 SAP HANA LandscapeHostConfiguration

To check the status of an HANA database and to find out if the cluster should react, you can use the script *landscapeHostConfiguration.py* to be called as Linux user *<sid>adm*.

```
suse01:~> HDBSettings.sh landscapeHostConfiguration.py
| Host   | Host   | ... NameServer   | NameServer  | IndexServer | IndexServer |
|        | Active | ... Config Role  | Actual Role | Config Role | Actual Role |
| ------ | ------ | ... ------------ | ----------- | ----------- | ----------- |
| suse01 | yes    | ... master 1     | master      | worker      | master      |

overall host status: ok
```

Following the SAP HA guideline, the SAP HANA resource agent interprets the return codes in the following way:

TABLE 6: INTERPRETATION OF RETURN CODES

Return Code	Interpretation
4	SAP HANA database is up and OK. The cluster does interpret this as a correctly running database.
3	SAP HANA database is up and in status info. The cluster does interpret this as a correctly running database.
2	SAP HANA database is up and in status warning. The cluster does interpret this as a correctly running database.
1	SAP HANA database is down. If the database should be up and is not down by intention, this could trigger a takeover.
0	Internal Script Error – to be ignored.

11.3 Maintenance

To receive updates for the operating system or SUSE Linux Enterprise High Availability, it is recommended to register your systems to either a local SUSE Manager, to Repository Mirroring Tool (RMT), or remotely with SUSE Customer Center. For more information, visit

the respective Web pages: https://www.suse.com/products/suse-manager/ https://documentation.suse.com/sles/15-SP6/html/SLES-all/book-rmt.html https://scc.suse.com/docs/help Examples for maintenance tasks are also given in manual page SAPHanaSR_maintenance_examples(7).

11.3.1 Updating the operating system and cluster

For an update of SUSE Linux Enterprise Server for SAP Applications packages including cluster software, follow the rolling update procedure defined in the product documentation of the SUSE Linux Enterprise High Availability Administration Guide, chapter *Upgrading Your Cluster and Updating Software Packages* at https://documentation.suse.com/sle-ha/15-SP6/single-html/SLE-HA-administration/#cha-ha-migration.

11.3.2 Updating SAP HANA - seamless SAP HANA maintenance

For updating SAP HANA database systems in system replication, you need to follow the defined SAP processes. This section describes the steps required before and after the update procedure to get the system replication automated again.

SUSE has optimized the SAP HANA maintenance process in the cluster. The improved procedure only sets the multi-state resource to maintenance and keeps the rest of the cluster (SAPHanaTopology clones and IPaddr2 vIP resource) still active. Using the updated procedure allows a seamless SAP HANA maintenance in the cluster, as the virtual IP address can automatically follow the running primary.

Prepare the cluster not to react on the maintenance work to be done on the SAP HANA database systems. Set the multi-state resource to `maintenance`.

EXAMPLE 31: MAIN SAP HANA UPDATE PROCEDURE

Pre-Update Tasks

For the multi-state-resource set the maintenance mode as follows:

```
# crm resource maintenance <multi-state-resource>
```

The <multi-state-resource> in the guide at hand is `mst_SAPHanaCon_HA1_HDB10`.

Update

Process the SAP Update for both SAP HANA database systems. This procedure is described by SAP.

Post-Update Tasks

Expect the primary/secondary roles to be exchanged after the maintenance. Therefore, tell the cluster to forget about these states and to reprobe the updated SAP HANA database systems.

```
# crm resource refresh <multi-state-resource>
```

After the SAP HANA update is complete on both sites, tell the cluster about the end of the maintenance process. This allows the cluster to actively control and monitor the SAP again.

```
# crm resource maintenance <multi-state-resource> off
```

Optionally, you could completety remove the maintenance attribute from the resource. For more details see manual page SAPHanaSR_maintenance_examples(7) and crm(8).

11.3.3 Migrating an SAP HANA primary

In the following procedures, we assume the primary runs on node 1 and the secondary on node 2. The goal is to "exchange" the roles of the nodes: the primary should then run on node 2 and the secondary should run on node 1.

There are different methods to get the exchange of the roles done. The following procedure shows how to tell the cluster to "accept" a role change via native SAP HANA commands.

EXAMPLE 32: MIGRATING AN SAP HANA PRIMARY USING SAP TOOLSET

Pre-Migration Tasks

Set the multi-state resource to `maintenance`. This can be done on any cluster node.

```
# crm resource maintenance <multi-state-resource>
```

Manual Takeover Process

- Stop the primary SAP HANA database system. Enter the command in our example on node1 as user *<sid>adm*.

  ```
  ~> HDB stop
  ```

- Before proceeding, make sure the primary SAP HANA database is stopped.
- Start the takeover process on the secondary SAP HANA database system. Enter the command in our example on node 2 as user *<sid>adm*.

```
~> hdbnsutil -sr_takeover
```

- Register the former primary to become the new secondary. Enter the command in our example on node1 as user *<sid>adm*.

```
~> hdbnsutil -sr_register --remoteHost=suse02 --remoteInstance=10 \
 --replicationMode=sync --name=WDF \
 --operationMode=logreplay
```

- Start the new secondary SAP HANA database system. Enter the command in our example on node1 as user *<sid>adm*.

```
~> HDB start
```

Post-Migration Tasks

- Wait some time until `SAPHanaSR-showAttr` shows both SAP HANA database systems to be up again (field roles must start with the digit 4). The new secondary should have role "S" (for secondary).
- Tell the cluster to forget about the former multi-state roles and to re-monitor the failed master. The command can be submitted on any cluster node as user root.

```
# crm resource refresh <multi-state-resource>
```

- Set the multi-state resource to the status managed again. The command can be submitted on any cluster node as user root.

```
# crm resource maintenance <multi-state-resource> off
```

The following paragraphs explain how to use the cluster to partially automate the migration.

EXAMPLE 33: MOVING AN SAP HANA PRIMARY USING THE CLUSTER TOOLSET

- Create a "move away" from this node rule by using the **force** option.

```
# crm resource move <multi-state-resource> force
```

Because of the "move away" (**force**) rule, the cluster will **stop** the current primary. After that, run a **promote** on the secondary site if the system replication was in sync before. You should not migrate the primary if the status of the system replication is not in sync (SFAIL).

Important

Migration without the **force** option will cause a takeover without the former primary to be stopped. Only the migration with **force** option is supported.

- Wait until the secondary has completely taken over to be the new primary role. You see this using the command line tool `SAPHanaSR-showAttr`. Now check for the attributes "roles" for the new primary. It must start with "**4:P**".

```
{sapnode1}:~ # SAPHanaSR-showAttr --format=script | grep "/roles="
Mon Jun 21 19:38:50 2021; Hosts/{sapnode1}/roles=*1:P*:master1::worker:
Mon Jun 21 19:38:50 2021; Hosts/{sapnode2}/roles=*4:P*:master1:master:worker:master
```

- If you have set up the parameter value `AUTOMATED_REGISTER="true"`, you can skip this step. In other cases you now need to register the old primary. Enter the command in our example on node1 as user *<sid>adm*.

```
~> hdbnsutil -sr_register --remoteHost=suse02 --remoteInstance=10 \
    --replicationMode=sync --operationMode=logreplay \
    --name=WDF
```

- Clear the ban rules of the resource to allow the cluster to start the new secondary.

```
# crm resource clear <multi-state-resource>
```

- Wait until the new secondary has started. You see this using the command line tool `SAPHanaSR-showAttr` and check for the attributes "roles" for the new primary. It must start with "**4:S**".

```
{sapnode1}:~ # SAPHanaSR-showAttr --format=script | grep "/roles="
Mon Jun 21 19:38:50 2021; Hosts/{sapnode1}/roles=*4:S*:master1::worker:
Mon Jun 21 19:38:50 2021; Hosts/{sapnode2}/roles=*4:P*:master1:master:worker:master
```

12 Examples

12.1 Example `ha-cluster-init` configuration

```
suse01:~ # ha-cluster-init -u
```

```
  Generating SSH key
  Configuring csync2
  Generating csync2 shared key (this may take a while)...done
  csync2 checking files...done

Configure Corosync (unicast):
  This will configure the cluster messaging layer.  You will need
  to specify a network address over which to communicate (default
  is eth0's network, but you can use the network address of any
  active interface).

  Address for ring0 [192.168.1.11]
  Port for ring0 [5405]

Configure SBD:
  If you have shared storage, for example a SAN or iSCSI target,
  you can use it avoid split-brain scenarios by configuring SBD.
  This requires a 1 MB partition, accessible to all nodes in the
  cluster.  The device path must be persistent and consistent
  across all nodes in the cluster, so /dev/disk/by-id/* devices
  are a good choice.  Note that all data on the partition you
  specify here will be destroyed.

Do you wish to use SBD (y/n)? y
  Path to storage device (e.g. /dev/disk/by-id/...), or "none" []/dev/disk/by-id/SBDA
WARNING: All data on /dev/disk/by-id/SBDA will be destroyed!
Are you sure you wish to use this device (y/n)? y
  Initializing SBD......done
  Hawk cluster interface is now running. To see cluster status, open:
    https://192.168.1.11:7630/
  Log in with username 'hacluster', password 'linux'
You should change the hacluster password to something more secure!
  Waiting for cluster........done
  Loading initial cluster configuration

Configure Administration IP Address:
  Optionally configure an administration virtual IP
  address. The purpose of this IP address is to
  provide a single IP that can be used to interact
  with the cluster, rather than using the IP address
  of any specific cluster node.

Do you wish to configure a virtual IP address (y/n)? n
  Done (log saved to /var/log/ha-cluster-bootstrap.log)
```

12.2 Example cluster configuration

The following complete crm configuration is for a two-node cluster (suse01, suse02) and an SAP HANA database with SID HA1 and instance number 10. Priority fencing prefers the SAP HANA primary in case of split-brain. The virtual IP address in the example is 192.168.1.20.

```
node suse01
node suse02

primitive rsc_SAPHanaTop_HA1_HDB10 ocf:suse:SAPHanaTopology \
    op start interval=0 timeout=600 \
    op stop interval=0 timeout=600 \
    op monitor interval=50 timeout=600 \
    params SID=HA1 InstanceNumber=10

primitive rsc_SAPHanaFil_HA1_HDB10 ocf:suse:SAPHanaFilesystem \
    op start interval=0 timeout=10 \
    op stop interval=0 timeout=20 \
    op monitor interval=120 timeout=120 ON_FAIL_ACTION="fence" \
    params SID=HA1 InstanceNumber=10

primitive rsc_SAPHanaCon_HA1_HDB10 ocf:suse:SAPHana \
    op start interval=0 timeout=3600 \
    op stop interval=0 timeout=3600 \
    op promote interval=0 timeout=900 \
    op demote interval=0 timeout=320 \
    op monitor interval=60 role=Promoted timeout=700 \
    op monitor interval=61 role=Unpromoted timeout=700 \
    params SID=HA1 InstanceNumber=10 PREFER_SITE_TAKEOVER=true \
           DUPLICATE_PRIMARY_TIMEOUT=7200 AUTOMATED_REGISTER=false \
    meta priority=100

primitive rsc_ip_HA1_HDB10 ocf:heartbeat:IPaddr2 \
    op monitor interval=10 timeout=20 \
    params ip=192.168.1.20

primitive stonith-sbd stonith:external/sbd \
    params pcmk_delay_max=15

clone mst_SAPHanaCon_HA1_HDB10 rsc_SAPHanaCon_HA1_HDB10 \
    meta clone-node-max=1 promotable=true interleave=true

clone cln_SAPHanaTop_HA1_HDB10 rsc_SAPHanaTop_HA1_HDB10 \
    meta clone-node-max=1 interleave=true

clone cln_SAPHanaFil_HA1_HDB10 rsc_SAPHanaFil_HA1_HDB10 \
```

```
    meta clone-node-max=1 interleave=true

colocation col_saphana_ip_HA1_HDB10 2000: \
    rsc_ip_HA1_HDB10:Started mst_SAPHanaCon_HA1_HDB10:Promoted
order ord_saphana_HA1_HDB10 Optional: \
    cln_SAPHanaTop_HA1_HDB10 mst_SAPHanaCon_HA1_HDB10

property cib-bootstrap-options: \
    cluster-infrastructure=corosync \
    stonith-enabled=true \
    stonith-action=reboot \
    stonith-timeout=150 \
    priority-fencing-delay=30

rsc_defaults rsc-options: \
    resource-stickiness=1000 \
    migration-threshold=5000

op_defaults op-options \
    timeout=600 \
    record-pending=true
```

12.3 Example for */etc/corosync/corosync.conf*

The following file shows a typical corosync configuration with two rings. Review the SUSE product documentation about details. See also manual pages corosync.conf(5) and votequorum(5).

```
# Read the corosync.conf.5 manual page
totem {
    version: 2
    secauth: on
    crypto_hash: sha1
    crypto_cipher: aes256
    cluster_name: suse-ha
    clear_node_high_bit: yes
    token: 5000
    token_retransmits_before_loss_const: 10
    join: 60
    consensus: 6000
    max_messages: 20
    interface {
        ringnumber: 0
        mcastport: 5405
        ttl: 1
    }
```

```
    interface {
        ringnumber: 1
        mcastport: 5407
        ttl: 1
    }
    rrp_mode: passive
    transport: udpu
}

logging {
    fileline: off
    to_stderr: no
    to_logfile: no
    logfile: /var/log/cluster/corosync.log
    to_syslog: yes
    debug: off
    timestamp: on
    logger_subsys {
        subsys: QUORUM
        debug: off
    }
}

nodelist {
    node {
            ring0_addr: 192.168.1.11
            ring1_addr: 192.168.2.11
            nodeid: 1
    }
    node {
            ring0_addr: 192.168.1.12
            ring1_addr: 192.168.2.12
            nodeid: 2
    }
}

quorum {
        provider: corosync_votequorum
        expected_votes: 2
        two_node: 1
}
```

12.4 Examples for alternate STONITH methods

12.4.1 Example for deterministic SBD STONITH

These SBD resources make sure that node suse01 will win in case of split-brain.

```
primitive rsc_sbd_suse01 stonith:external/sbd \
    params pcmk_host_list=suse02 pcmk_delay_base=0

primitive rsc_sbd_suse02 stonith:external/sbd \
    params pcmk_host_list=suse01 pcmk_delay_base=30
```

12.4.2 Example for the IPMI STONITH method

```
primitive rsc_suse01_stonith stonith:external/ipmi \
    params hostname="suse01" ipaddr="192.168.1.101" userid="stonith" \
    passwd="k1llm3" interface="lanplus" \
    op monitor interval=1800 timeout=30
    ...
primitive rsc_suse02_stonith stonith:external/ipmi \
    params hostname="suse02" ipaddr="192.168.1.102" userid="stonith" \
    passwd="k1llm3" interface="lanplus" \
    op monitor interval=1800 timeout=30
    ...
location loc_suse01_stonith rsc_suse01_stonith -inf: suse01
location loc_suse02_stonith rsc_suse02_stonith -inf: suse02
```

12.5 Example for checking legacy SystemV integration

Check if the SAP hostagent is installed on all cluster nodes. As Linux user *root*, use the commands `systemctl` and `saphostctrl` to check the SAP hostagent:

```
# systemctl status sapinit
* sapinit.service - LSB: Start the sapstartsrv
   Loaded: loaded (/etc/init.d/sapinit; generated; vendor preset: disabled)
   Active: active (exited) since Wed 2022-02-09 17:25:36 CET; 3 weeks 0 days ago
     Docs: man:systemd-sysv-generator(8)
    Tasks: 0
   CGroup: /system.slice/sapinit.service
# /usr/sap/hostctrl/exe/saphostctrl -function ListInstances
Inst Info : HA1 - 10 - suse01 - 753, patch 819, changelist 2069355
```

The SystemV style sapinit is running and the hostagent recognises the installed database.

As Linux user *<sid>adm*, use the command line tool `HDB` to get an overview of running SAP HANA processes. The output of `HDB info` should be similar to the output shown below:

```
suse01:ha1adm> HDB info
USER          PID     PPID  ... COMMAND
ha1adm      13017    ... -sh
ha1adm      13072    ...  \_ /bin/sh /usr/sap/HA1/HDB10/HDB info
ha1adm      13103    ...      \_ ps fx -U ha1adm -o
 user:8,pid:8,ppid:8,pcpu:5,vsz:10,rss:10,args
ha1adm       9268    ... hdbrsutil  --start --port 31003 --volume 2 --volumesuffix
 mnt00001/hdb00002.00003 --identifier 1580897137
ha1adm       8911    ... hdbrsutil  --start --port 31001 --volume 1 --volumesuffix
 mnt00001/hdb00001 --identifier 1580897100
ha1adm       8729    ... sapstart pf=/hana/shared/HA1/profile/HA1_HDB10_suse01
ha1adm       8738    ...  \_ /usr/sap/HA1/HDB10/suse01/trace/hdb.sapHA1_HDB10 -d -nw -f /
usr/sap/HA1/HDB10/suse01/daemon.ini pf=/usr/sap/HA1/SYS/profile/HA1_HDB10_suse01
ha1adm       8756    ...      \_ hdbnameserver
ha1adm       9031    ...      \_ hdbcompileserver
ha1adm       9034    ...      \_ hdbpreprocessor
ha1adm       9081    ...      \_ hdbindexserver -port 31003
ha1adm       9084    ...      \_ hdbxsengine -port 31007
ha1adm       9531    ...      \_ hdbwebdispatcher
ha1adm       8574    ... /usr/sap/HA1/HDB10/exe/sapstartsrv pf=/hana/shared/HA1/profile/
HA1_HDB10_suse01 -D -u ha1adm
```

13 References

For more detailed information, have a look at the documents listed below.

13.1 SUSE Product Documentation

Best Practices for SAP on SUSE Linux Enterprise

https://documentation.suse.com/sbp/sap/

SUSE product manuals and documentation

https://documentation.suse.com/

Release notes

https://www.suse.com/releasenotes/

Online documentation of SLES for SAP

https://documentation.suse.com/sles-sap/15-SP4/

Online documentation of SUSE Linux Enterprise High Availability

https://documentation.suse.com/sle-ha/15-SP6/single-html/SLE-HA-administration/

Deployment guide for SUSE Linux Enterprise Server

https://documentation.suse.com/sles/15-SP6/single-html/SLES-deployment/

Tuning guide for SUSE Linux Enterprise Server

https://documentation.suse.com/sbp/all/single-html/SBP-performance-tuning/

Storage administration guide for SUSE Linux Enterprise Server

https://documentation.suse.com/sles/15-SP6/single-html/SLES-storage/

SUSE Linux Enterprise Server Persistent Memory Guide

https://documentation.suse.com/sles/15-SP6/html/SLES-all/cha-nvdimm.html

SUSE Linux Enterprise kernel specs

https://www.suse.com/releasenotes/x86_64/SUSE-SLES/15-SP4/index.html#kernel-limits

SUSE Linux Enterprise file system specs

https://www.suse.com/releasenotes/x86_64/SUSE-SLES/15-SP4/index.html#file-system-comparison

XFS file system

https://www.suse.com/c/xfs-the-file-system-of-choice/

SUSE YES certified hardware database

https://www.suse.com/yessearch/

SUSE Manager Product Page

https://www.suse.com/products/suse-manager/

SUSE Manager Documentation

https://documentation.suse.com/external-tree/en-us/suma/4.1/suse-manager/index.html

RMT = Repository Mirroring Tool documentation

https://documentation.suse.com/sles/15-SP6/html/SLES-all/book-rmt.html

SUSE Customer Center Fequently Asked Questions

https://scc.suse.com/docs/help

13.2 Related Manual Pages

- chronyc(8)
- corosync.conf(8)
- corosync_overview(8)
- crm(8)
- crm_mon(8)
- crm_simulate(8)
- cs_clusterstate(8)
- cs_man2pdf(8)
- cs_show_hana_info(8)
- cs_show_sbd_devices(8)
- cs_wait_for_idle(8)
- ha_related_sap_notes(7)
- ha_related_suse_tids(7)
- ocf_heartbeat_IPaddr2(7)

- ocf_heartbeat_SAPInstance(7)
- ocf_suse_SAPHanaController(7)
- ocf_suse_SAPHanaFilesystem(7)
- ocf_suse_SAPHanaTopology(7)
- ocf_suse_SAPHanaFilesystem(7)
- SAPHanaSR(7)
- SAPHanaSR-alert-fencing(8)
- SAPHanaSR-angi(7)
- SAPHanaSR_basic_cluster(7)
- SAPHanaSR-hookHelper(8)
- SAPHanaSR_maintenance_examples(7)
- SAPHanaSR-manageAttr(8)
- SAPHanaSR-manageProvider(8)
- SAPHanaSR-monitor(8)
- SAPHanaSR-replay-archive(8)
- SAPHanaSR-ScaleOut(7)
- SAPHanaSR-ScaleOut_basic_cluster(7)
- SAPHanaSR-showAttr(8)
- SAPHanaSR-show-hadr-runtimes(8)
- saptune(8)
- sbd(8)
- stonith_sbd(7)
- sudo(8)
- sudoers(5)
- supportconfig(8)

- susChkSrv.py(7)
- susCostOpt.py(7)
- susHanaSR.py(7)
- susTkOver.py(7)
- systemctl(8)
- systemd-cgls(8)
- votequorum(5)
- zypper(8)

13.3 Related SUSE TIDs

SAP HANA SR Performance Optimized Scenario - Setup Guide - Errata

https://www.suse.com/support/kb/doc/?id=7023882

Estimate correct multipath timeout

https://www.suse.com/support/kb/doc/?id=7016305

Can't open watchdog device: /dev/watchdog: Device or resource busy

https://www.suse.com/support/kb/doc/?id=7008216

Systemd-udev-settle timing out

https://www.suse.com/support/kb/doc/?id=7022681

Configuring Persistent Memory Devices (PMEM) results in booting to the recovery shell

https://www.suse.com/support/kb/doc/?id=000019517

Slow boot boot initialization on machines with Intel Optane DC Memory causing auto-mount to fail

https://www.suse.com/support/kb/doc/?id=000019462

How to load the correct watchdog kernel module

https://www.suse.com/support/kb/doc/?id=7016880

TID XFS metadata corruption and invalid checksum on SAP Hana servers

https://www.suse.com/support/kb/doc/?id=7022921

Overcommit Memory in SLES

https://www.suse.com/support/kb/doc/?id=7002775

Recommended SUSE SLES 4 SAP Settings

https://www.suse.com/support/kb/doc/?id=7024082

SAPHanaController running in timeout when starting SAP Hana

https://www.suse.com/support/kb/doc/?id=000019899

Troubleshooting the SAPHanaSR python hook

https://www.suse.com/support/kb/doc/?id=000019865

Entry "CALLING CRM: ... rc=256" in HANA trace after upgrading SAPHanaSR-ScaleOut

https://www.suse.com/support/kb/doc/?id=000020599

SAP HANA monitors timed out after 5 seconds

https://www.suse.com/support/kb/doc/?id=000020626

HA cluster takeover takes too long on HANA indexserver failure

https://www.suse.com/support/kb/doc/?id=000020845

Cluster node fence as SAPHanaTopology fails with error code 1 (OCF_ERR_GENERIC) during a normal cluster stop

https://www.suse.com/support/kb/doc/?id=000020964

SUSE HA for HANA cluster node fenced at shutdown, despite of systemd integration

https://www.suse.com/support/kb/doc/?id=000021046

SAP HANA scale-out - pacemaker.service: "Cannot find sapstartsrv and sapcontrol executable, please set DIR_EXECUTABLE parameter!"

https://www.suse.com/support/kb/doc/?id=000021062

SAPHanaSR-showAttr fails with error "Error: NIECONN_REFUSED ..."

https://www.suse.com/support/kb/doc/?id=000020548

Protect HANA against manually caused dual-primary situation in SUSE HA cluster

https://www.suse.com/support/kb/doc/?id=000021044

Address space monitoring and HANA DB performance

https://www.suse.com/support/kb/doc/?id=000020746

HANA DB resource failed to start

https://www.suse.com/support/kb/doc/?id=000020948

SAPHanaController monitor timeout leads to database restart

https://www.suse.com/support/kb/doc/?id=000021249

HANA Database Planning Engine crashes in __strncmp_avx2_rtm+0x1b3

https://www.suse.com/support/kb/doc/?id=000021026

Basic health check for two-node SAP HANA performance based model

https://www.suse.com/support/kb/doc/?id=7022984

How to re-enable replication in a two-node SAP performance based model

https://www.suse.com/support/kb/doc/?id=7023127

Showing SOK Status in Cluster Monitoring Tools Workaround

https://www.suse.com/support/kb/doc/?id=7023526

HANA SystemReplication doesn’t provide SiteName to Corosync Cluster

https://www.suse.com/support/kb/doc/?id=000019754

SUSE Cluster Support for SAP HANA System Replication Active / Active Read Enabled Feature

https://www.suse.com/support/kb/doc/?id=7023884

SAP Generating 'Database host operating system is not supported' alerts

https://www.suse.com/support/kb/doc/?id=7023744

sapstartsrv does not respawn after a forceful kill of the master nameserver

https://www.suse.com/support/kb/doc/?id=7024291

SAPHanaSR HANA system replication automation without layer 2 network

https://www.suse.com/support/kb/doc/?id=000020333

The vIP cluster resource does not follow the SAP HANA master ...

https://www.suse.com/support/kb/doc/?id=000019769

Handling failed NFS share in SUSE HA cluster for HANA system replication

https://www.suse.com/support/kb/doc/?id=000019904

SAP Instances failed stop on shutdown (PACEMAKER, SYSTEMD, SAP)

https://www.suse.com/support/kb/doc/?id=7022671

SAP on SLES shows Error: NIECONN_REFUSED in the logs

https://www.suse.com/support/kb/doc/?id=7023236

Indepth HANA Cluster Debug Data Collection (PACEMAKER, SAP)

https://www.suse.com/support/kb/doc/?id=7022702

How to prevent certain values in limits.conf from being changed by saptune

https://www.suse.com/support/kb/doc/?id=7023104

Disabling fstrim - under which conditions?

https://www.suse.com/support/kb/doc/?id=7023805

saptune: WARNING saptune.io.go:66: 'noop' is not a valid scheduler for device

https://www.suse.com/support/kb/doc/?id=000019572

How to patch a SAP Application Pacemaker Cluster

https://www.suse.com/support/kb/doc/?id=000020268

13.4 Related SUSE blogs

How to upgrade to SAPHanaSR-angi

https://www.suse.com/c/how-to-upgrade-to-saphanasr-angi/

Emergency Braking for SAP HANA Dying Indexserver

https://www.suse.com/c/emergency-braking-for-sap-hana-dying-indexserver/

SAP HANA Cockpit with SUSE HA integration greatly improves data integrity

https://www.suse.com/c/sap-hana-cockpit-with-suse-ha-integration-greatly-improves-data-integrity/

Handover for the Next Round – SAP on SUSE Cluster and systemd Native Integration

https://www.suse.com/c/handover-for-the-next-round-sap-on-suse-cluster-and-systemd-native-integration/

SAPHanaSR-ScaleOut for Multi-Target Architecture and Principles

https://www.suse.com/c/saphanasr-scaleout-multi-target/

SAP HANA Scale-Out System Replication for large ERP Systems

https://www.suse.com/c/sap-hana-scale-out-system-replication-for-large-erp-systems/

SAP HANA Cost-optimized – An alternative Route is available

https://www.suse.com/c/sap-hana-cost-optimized-an-alternative-route-is-available/

Let's flip the flags! Is my SAP HANA database in sync or not?

https://www.suse.com/c/lets-flip-the-flags-is-my-sap-hana-database-in-sync-or-not/

Entry to blog series #towardsZeroDowntime

https://www.suse.com/c/tag/towardszerodowntime/

Fail-Safe Operation of SAP HANA: SUSE Extends Its High-Availability Solution

http://scn.sap.com/community/hana-in-memory/blog/2014/04/04/fail-safe-operation-of-sap-hana-suse-extends-its-high-availability-solution

13.5 Related SAP Documentation

SAP Product Availability Matrix

https://support.sap.com/en/release-upgrade-maintenance.html#section_1969201630

SAP HANA Installation and Update Guide

https://help.sap.com/doc/e9702d76c3284623b02de196c0e79e49/2.0.05/en-US/SAP_HANA_Server_Installation_Guide_en.pdf

SAP HANA Administration Guide

https://help.sap.com/doc/eb75509ab0fd1014a2c6ba9b6d252832/2.0.05/en-US/SAP_HANA_Administration_Guide_en.pdf

SAP HANA Documentation Entry Page

https://help.sap.com/viewer/product/SAP_HANA_PLATFORM/2.0.05/en-US

SAP HANA Tailored Data Center Integration - FAQ

https://www.sap.com/documents/2016/05/e8705aae-717c-0010-82c7-eda71af511fa.html

SAP HANA and Persistent Memory

https://blogs.sap.com/2020/01/30/sap-hana-and-persistent-memory/

SAP HANA HA/DR Provider Hook Methods

https://help.sap.com/viewer/6b94445c94ae495c83a19646e7c3fd56/2.0.05/en-US/5df2e766549a405e95de4c5d7f2efc2d.html

13.6 Related SAP Notes

611361 - Hostnames of SAP servers

https://launchpad.support.sap.com/#/notes/611361

768727 - Automatic restart functions in sapstart for processes

https://launchpad.support.sap.com/#/notes/768727

927637 - Web service authentication in sapstartsrv as of Release 7.00

https://launchpad.support.sap.com/#/notes/927637

1092448 - IBM XL C/C++ runtime environment for Linux on system p

https://launchpad.support.sap.com/#/notes/1092448

1514967 - SAP HANA: Central Note

https://launchpad.support.sap.com/#/notes/1514967

1552925 - Linux: High Availability Cluster Solutions

https://launchpad.support.sap.com/#/notes/1552925

1763512 - Support details for SUSE Linux Enterprise for SAP Applications

https://launchpad.support.sap.com/#/notes/1763512

1846872 - "No space left on device" error reported from HANA

https://launchpad.support.sap.com/#/notes/1846872

1876398 - Network configuration for System Replication in HANA SP6

https://launchpad.support.sap.com/#/notes/1876398

1888072 - SAP HANA DB: Indexserver crash in strcmp sse42

https://launchpad.support.sap.com/#/notes/1888072

2021789 - SAP HANA Revison and Maintenance Strategy

https://launchpad.support.sap.com/#/notes/2021789

2196941 - SAP HANA Software Replication Takeover Hook Changes

https://launchpad.support.sap.com/#/notes/2196941

2235581 - SAP HANA: Supported Operating Systems

https://launchpad.support.sap.com/#/notes/2235581

2369981 - Required configuration steps for authentication with HANA System Replication

https://launchpad.support.sap.com/#/notes/2369981

2369910 - SAP Software on Linux: General information

https://launchpad.support.sap.com/#/notes/2369910

2380229 - SAP HANA Platform 2.0 - Central Note

https://launchpad.support.sap.com/#/notes/2380229

2434562 - System Replication Hanging in Status "SYNCING" or "ERROR" With Status Detail "Missing Log" or "Invalid backup size"

https://launchpad.support.sap.com/#/notes/2434562

2578899 - SUSE Linux Enterprise Server 15: Installation Note

https://launchpad.support.sap.com/#/notes/2578899

2647673 - HANA Installation Failure

https://launchpad.support.sap.com/#/notes/2647673

2684254 - SAP HANA DB: Recommended OS settings for SLES 15 / SLES for SAP Applications 15

https://launchpad.support.sap.com/#/notes/2684254

2733483 - Host Auto-Failover Not Occur when Indexserver Crash on Worker Node

https://launchpad.support.sap.com/#/notes/2733483

2750199 - Incorrect Alert Regarding Unsupported Operating System Version

https://launchpad.support.sap.com/#/notes/2750199

2844322 - SAP HANA Platform 2.0 SPS 05 Release Note

https://launchpad.support.sap.com/#/notes/2844322

2945239 - SAP HANA Platform 2.0 SPS 06 Release Note

https://launchpad.support.sap.com/#/notes/2945239

3007062 - FAQ: SAP HANA & Third Party Cluster Solutions

https://launchpad.support.sap.com/#/notes/3007062

3014176 - Applying System Size Dependent Resource Limits During Installation or Upgrade

https://launchpad.support.sap.com/#/notes/3014176

3043459 - SAP HANA 2 SPS05 Revision 056.00

https://launchpad.support.sap.com/#/notes/3043459

3072590 - Python 3 Support for Non-Productive SAP HANA Systems

https://launchpad.support.sap.com/#/notes/3072590

3070359 - Python 3 Migration Guide For SAP HANA

https://launchpad.support.sap.com/#/notes/3070359

3084229 - SAP HANA Python Support Scripts Fail due to Incompatibility With Python 3

https://launchpad.support.sap.com/#/notes/3084229

3091152 - sapstartsrv - improved deregistration for UNIX/Linux

https://launchpad.support.sap.com/#/notes/3091152

3093542 - Transition to Python 3 of the Python Distribution Delivered With SAP HANA 2.0 Server

https://launchpad.support.sap.com/#/notes/3093542

3139184 - Linux: systemd integration for sapstartsrv and SAP Hostagent

https://launchpad.support.sap.com/#/notes/3139184

3145200 - SAP Host Agent 7.22 PL57

https://launchpad.support.sap.com/#/notes/3145200

13.7 Pacemaker

Pacemaker Project Documentation

https://clusterlabs.org/pacemaker/doc/

14 Legal notice

15 GNU Free Documentation License

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document’s license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all

Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
   Permission is granted to copy, distribute and/or modify this document
   under the terms of the GNU Free Documentation License, Version 1.2
```

```
   or any later version published by the Free Software Foundation;
   with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
   A copy of the license is included in the section entitled “GNU
   Free Documentation License”.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “ with…Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
   Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.