

SAP HANA System Replication Scale-Up - Cost Optimized Scenario

SUSE Linux Enterprise Server for SAP Applications 15

Fabian Herschel, Distinguished Architect SAP, SUSE

Bernd Schubert, SAP Solution Architect, SUSE

Lars Pinne, System Engineer, SUSE

Thomas Korber, Linux Architect, B1 Systems GmbH

Eike Waldt, Linux Consultant & Trainer, B1 Systems GmbH

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 Scale-Up system replication automation in the cost optimized scenario. It is based on SUSE Linux Enterprise Server for SAP Applications 15 SP2. The concept however can also be used with newer versions.

Disclaimer: This document is part of the SUSE Best Practices series. All documents published in this series were contributed voluntarily by SUSE employees and by third parties. If not stated otherwise inside the document, the articles are intended only to be one example of how a particular action could be taken. Also, SUSE cannot verify either that the actions described in the articles do what they claim to do or that they do not have unintended consequences. All information found in this document has been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. Therefore, we need to specifically state that neither SUSE LLC, its affiliates, the authors, nor the translators may be held liable for possible errors or the consequences thereof.

Disclaimer: Documents published as part of the SUSE Best Practices series have been contributed voluntarily by SUSE employees and third parties. They are meant to serve as examples of how particular actions can be performed. They have been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. SUSE cannot verify that actions described in these documents do what is claimed or whether actions described have unintended consequences. SUSE LLC, its affiliates, the authors, and the translators may not be held liable for possible errors or the consequences thereof.

Publication Date: 2022-10-25

Contents

- 1 About this guide 4

2	Supported scenarios and prerequisites	13
3	Scope of this document	17
4	Planning the installation	19
5	Setting up the operating system	23
6	Installing the SAP HANA Databases on both cluster nodes	25
7	Setting up SAP HANA System Replication	27
8	Setting up SAP HANA HA/DR providers	32
9	Configuring the cluster	37
10	Testing the cluster	51
11	Administration	72
12	References	83
13	Examples	93
14	Legal notice	100
15	GNU Free Documentation License	101

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 cost optimized scenario**.

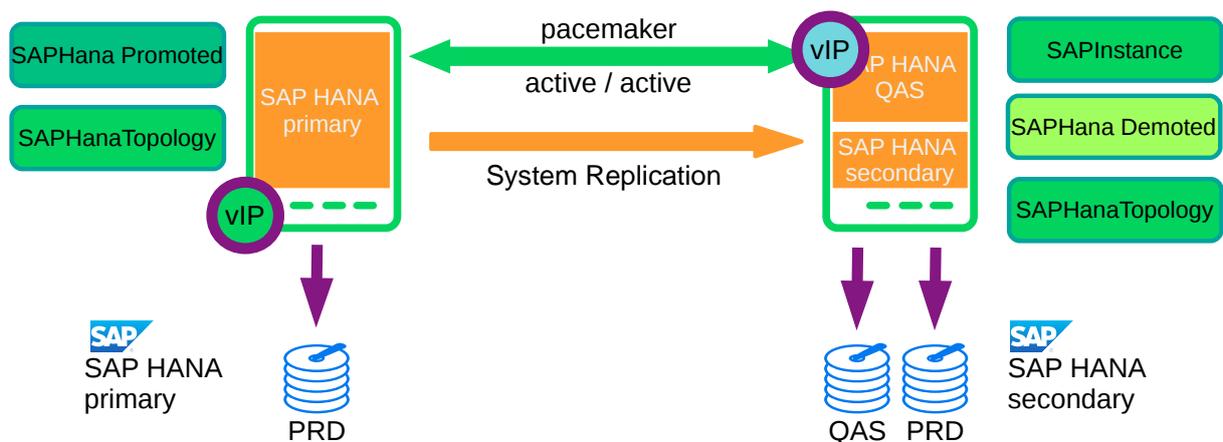


FIGURE 1: COST OPTIMIZED SCENARIO DESCRIBED IN THIS SETUP GUIDE

For an overview of supported scenarios, see [Section 1.1.3, "Scale-up scenarios and resource agents"](#) (page 7).

SAP HANA is the only database platform for prominent SAP platforms like SAP S/4HANA. SAP NetWeaver can also use SAP HANA as database back-end. As SAP HANA is only available on the Linux operating system, this triggers lots of Unix-to-Linux and Windows-to-Linux migrations.

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 two 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 Cost Optimized".

From the application perspective, the following variants are covered:

- plain system replication
- multi-tenant database containers

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 platforms (for example <https://www.suse.com/c/automating-the-sap-hana-high-availability-cluster-deployment-for-microsoft-azure/>). Ask your public cloud provider or your SUSE contact for details.

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

1.1.2 Scale-up versus scale-out

The first set of scenarios includes the architecture and development of *scale-up* solutions.

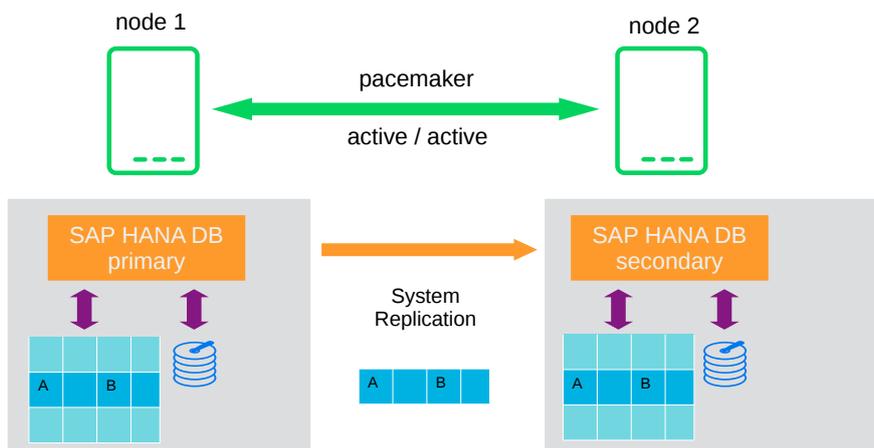


FIGURE 2: SAP HANA SYSTEM REPLICATION SCALE-UP IN THE CLUSTER

For these scenarios, SUSE has developed the scale-up resource agent package SAPHanaSR. 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 architecture and development of *scale-out* solutions (multi-box replication). For these scenarios, SUSE has developed the scale-out resource agent package SAPHanaSR-ScaleOut.

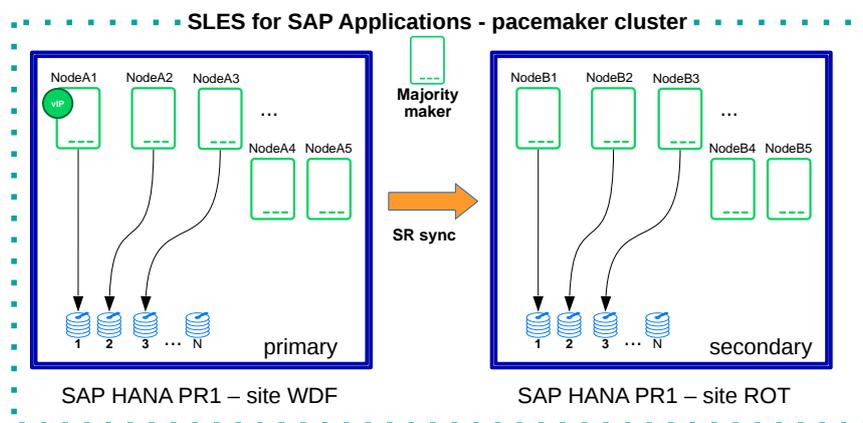


FIGURE 3: SAP HANA SYSTEM REPLICATION SCALE-OUT IN THE CLUSTER

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 SAPHana 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 another document available from the documentation Web page (<https://documentation.suse.com/sbp/sap/>). The document for *performance optimized* is named "SAP HANA System Replication Scale-Up - Performance Optimized Scenario".

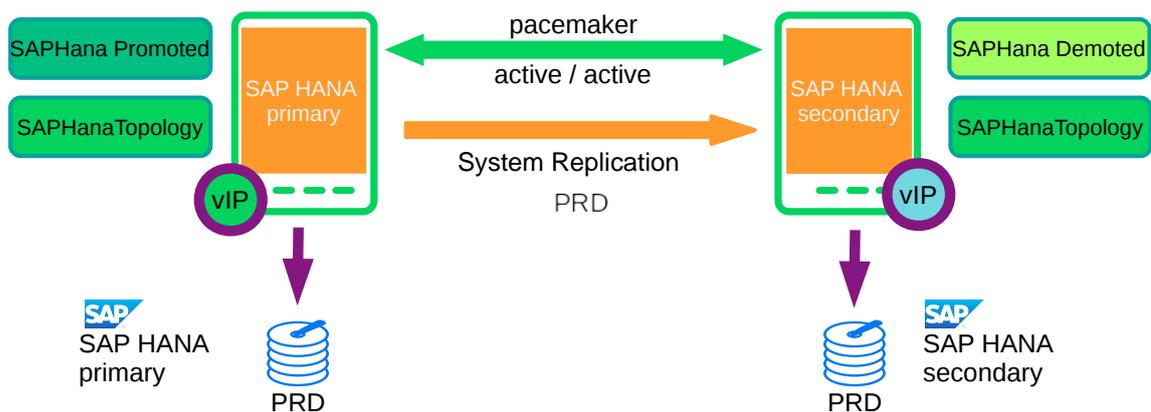


FIGURE 4: SAP HANA SYSTEM REPLICATION SCALE-UP IN THE CLUSTER - PERFORMANCE OPTIMIZED

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** ($A \Rightarrow B, Q$). This scenario and setup is described in this document.

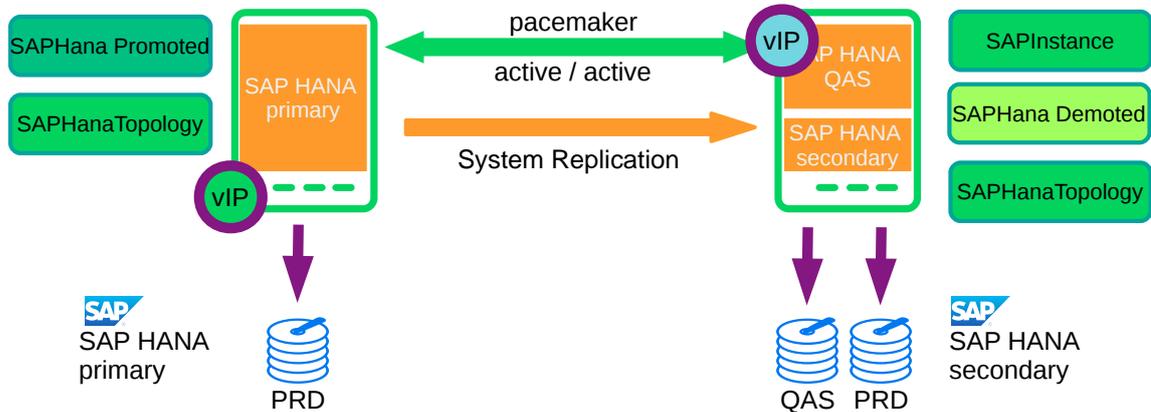


FIGURE 5: SAP HANA SYSTEM REPLICATION SCALE-UP IN THE CLUSTER - COST OPTIMIZED

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** ($A \Rightarrow B \rightarrow C$) and **Multi-target** ($B \leftarrow A \Rightarrow C$).

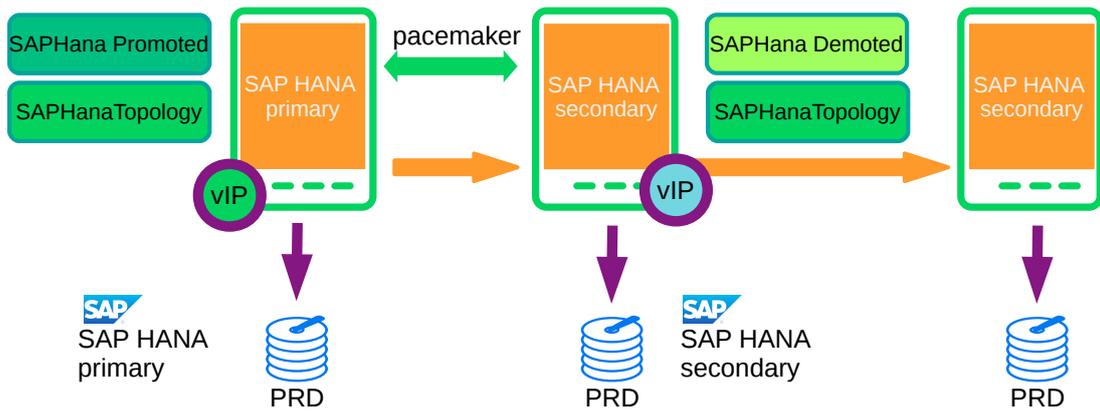


FIGURE 6: SAP HANA SYSTEM REPLICATION SCALE-UP IN THE CLUSTER - PERFORMANCE OPTIMIZED CHAIN

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.

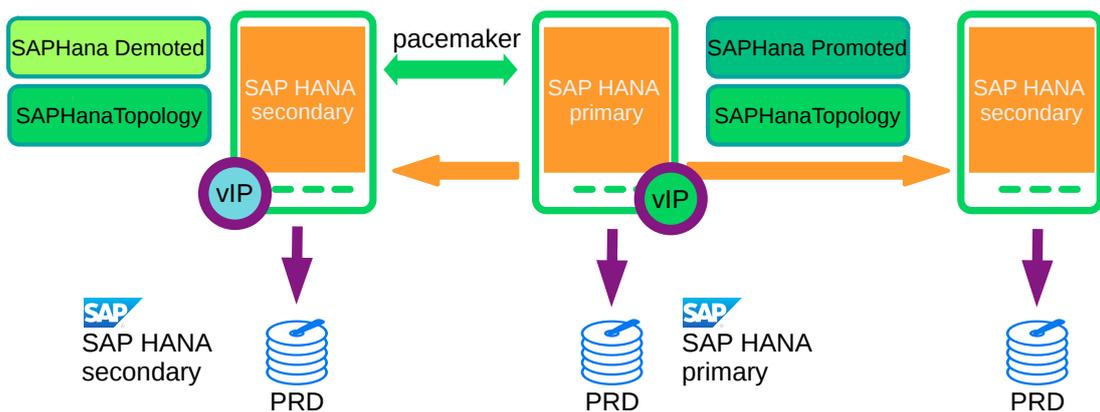


FIGURE 7: SAP HANA SYSTEM REPLICATION SCALE-UP IN THE CLUSTER - PERFORMANCE OPTIMIZED MULTI-TARGET

Multi-tier and multi-target systems are implemented as described in the document for the *performance optimized* scenario named "*SAP HANA System Replication Scale-Up - Performance Optimized Scenario*". Multi-tier and multi-target systems are only supported in the *performance optimized* scenario and *not with the cost optimized* scenario. They are mentioned here to give an overview of our entire portfolio of solutions.

In the multi-tier and multi-target scenario only the first replication pair (A and B) is handled by the Linux cluster.

- **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 cost optimized scenario

SAP allows to run a non-replicated instance of SAP HANA on the system replication site on the secondary site. Such a non-replicated database could be a development (DEV), test (TST), or quality assurance system (QAS).

In case of a failure of the primary SAP HANA on the primary site the cluster first tries to restart the failed SAP HANA database locally on this node. If the restart is not possible or if the complete primary node crashed, the takeover process will be triggered.

In case of a takeover, the secondary (replica) of this SAP HANA on node 2 is promoted **after** the shutdown of the non-replicated SAP HANA.

Alternatively you can configure a different resource handling procedure, but we recommend to try to restart SAP HANA locally first, as a takeover with non-preloaded tables can consume much time. Also, the needed graceful stop of the non-replicated system will take additional time. Thus, in many environments, the local restart will be faster.

To achieve an automation of this resource handling process, use the SAP HANA resource agents included in SAPHanaSR. System replication of the productive database is done using the resource agents SAPHana and SAPHanaTopology. The handling of the non-replicated database is implemented using the SAPInstance resource agent.

While SAPHana and SAPHanaTopology are driving the automation of the SAP HANA system replication, SAPInstance is used for the non-replicated SAP HANA database. In the past the architecture used SAPDatabase instead. The move to SAPInstance was needed to get rid of the error-prone user secure store keys setup procedure. Find more details in the blog article "SAP HANA Cost-optimized – An alternative Route is available" at <https://suse.com/c/sap-hana-cost-optimized-an-alternative-route-is-available/> ↗.

The automated shutdown of the non-replicated SAP HANA database (for example QAS) is achieved by cluster rules. More precisely, it is an anti-colocation of SAP HANA promoted versus SAP HANA non-replicated. This means if the primary SAP HANA system (like HA1) fails, the anti-colocation rules for the SAP HANA non-replicated system (like QAS) are triggered and the SAPInstance resource agent shuts down the non-replicated SAP HANA database.

The takeover to the secondary site takes up a lot of time, because the non-replicated database needs to be stopped gracefully prior to take over the productive database. This extended takeover time is the main disadvantage of the cost optimized scenario. Thus the cost optimized scenario might be combined with persistent memory to benefit from SAP HANA's persistent memory features.



Note

If you want to achieve a very fast takeover, the performance optimized scenario is the better option.

In addition to the description of the concept in this best practice document, read the corresponding SAP documentation such as "Using Secondary Servers for Non-Productive systems". The section is available for example for SAP HANA 2.0 SPS05 at <https://help.sap.com/viewer/6b94445c94ae495c83a19646e7c3fd56/2.0.05/en-US/5447545b91a04cf8a0d6133a026f2be5.html>.

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 'move' the primary or secondary instance using HAWK or any other cluster client commands. In *Section 11, "Administration"* 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/> .

Numerous whitepapers, best practices documents, setup guides, and other resources are available from the SUSE Best Practices Web page under the categories 'SAP Applications on SUSE Linux Enterprise' at <https://documentation.suse.com/sbp/sap/> .

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/> .

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:

- 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

With the SAPHanaSR resource agent software package, 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 Extension 15 (like SBD, IPMI) is supported with SAPHanaSR.
- 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 Extension 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 Extension 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 Extension about those recommendations.

- Automated registration of a failed primary after takeover prerequisites need to be defined.
 - As initial 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"`.
 - The cluster automates one single takeover in case of a failed primary. After that happened, the initial state needs to be restored by the administrative procedure outlined in this guide.
- 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 and multi-tier).
 - 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.
 - The scenario Multi-SID (MCOS) is not supported together with the cost optimized scenario. Hence, only one replicating database pair and one non-replicating database in the same cluster as described in this guide are supported for the cost-optimized scenario.

The SUSE Linux Enterprise Server for SAP Applications versions are:

- You need at least SAPHanaSR version 0.160 and at least SUSE Linux Enterprise Server for SAP Applications 15 SP2.
- Intel Optane DCPMM (aka PMEM) is supported since SUSE Linux Enterprise Server for SAP Applications 15 GA or newer.
- IBM Power vPMEM is supported since SUSE Linux Enterprise Server for SAP Applications 15 SP1 or newer.

For the HA/DR provider hook scripts SAPHanaSR.py and susCostOpt.py, the following requirements apply:

- SAP HANA 2.0 SPS04 or later provides the HA/DR provider hook method `srConnection-Chanegd()` with multi-target aware parameters. SAP HANA 1.0 does not provide them. The multi-target aware parameters are needed for the SAPHanaSR scale-up package.
- SAP HANA 2.0 SPS04 or later provides the HA/DR provider hook method `postTakeover()`.
- The user `<sid>adm` needs execution permission as user root for the command `crm_tribute`.

Additional considerations for the SAP HANA version are:

- Particularly the SAP HANA system replication cost-optimized scenario can benefit from SAP HANA's persistent memory features. Local restart and takeover are affected by the time SAP HANA needs for shutdown and loading column store. SAP HANA 2.0 SPS04 and later support persistent memory.

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.

In addition to SAP HANA, you need to install the SAP Host Agent on your system.

For information on supported hardware and virtualization, refer to the SUSE release notes and hardware compatibility database:

- <https://www.suse.com/releasenotes/>
- <https://www.suse.com/yessearch/>

Also, take a look at the SAP HANA product availability matrix, which can for example be found at https://support.sap.com/en/release-upgrade-maintenance.html#section_1969201630.

Additional information for deploying the cost optimized scenario in particular public clouds is available either from the respective cloud provider or from SUSE at <https://documentation.suse.com/sbp/sap/>.

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. In this document SUSE Linux Enterprise Server for SAP Applications 15 SP3 is used. This concept can also be used with SUSE Linux Enterprise Server for SAP Applications 15 SP2 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 SP3 systems. In addition, a non-replicated SAP HANA is installed and added to the cluster control.

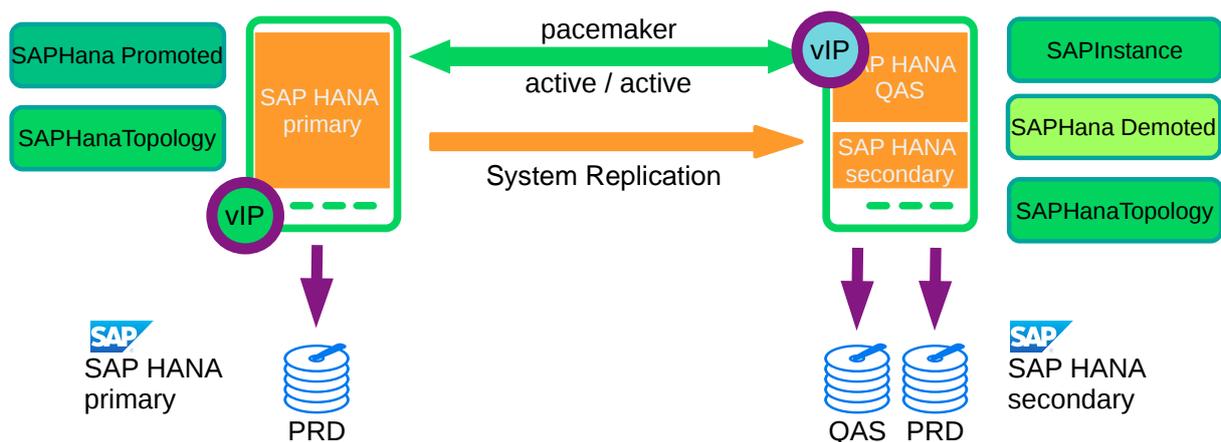


FIGURE 8: CLUSTER WITH SAP HANA SR - COST OPTIMIZED

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 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-SP3/html/SLES-SAP-guide/cha-cluster.html>.

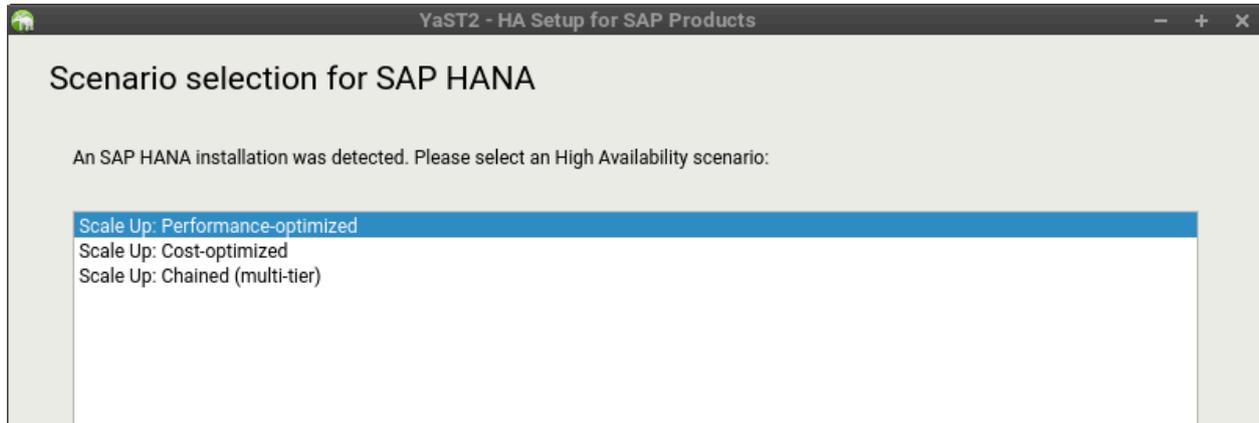
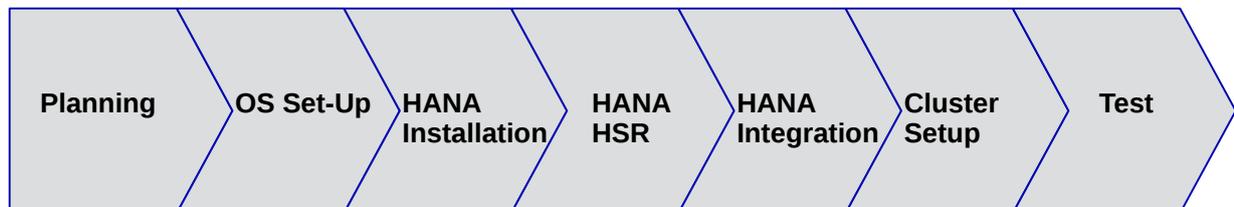


FIGURE 9: SCENARIO SELECTION FOR SAP HANA IN THE YAST MODULE SAP_HA

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.

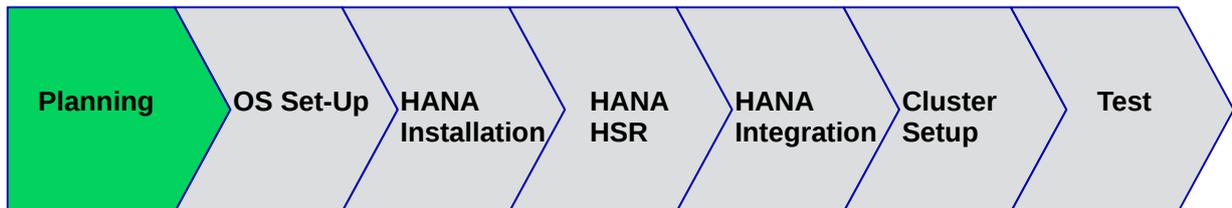
The seven main setup steps are:



- 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. The following minimum setup uses the half-size of RAM for the secondary SAP HANA database which has table preload inactive. 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):

- 1 VM with 32GB RAM, 50 GB disk space for the system
- 1 VM with 48GB RAM, 50 GB 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 96 GB for SAP HANA
- 1 data disk (for the non-replicated database) with a capacity of 96 GB for SAP HANA
- 1 additional IP address for takeover
- 1 additional IP address for non-replicated database
- 1 optional IP address for HAWK Administration GUI

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

- 1 VM with 32 GB RAM, 50 GB disk space for the system
- 1 VM with 48 GB RAM, 50 GB disk space for the system
- 1 VM with 2 GB RAM, 50 GB disk space for the system
- 2 data disks (one per site) with a capacity of each 96 GB for SAP HANA
- 1 data disk (for the non-replicated database) with a capacity of 96 GB for SAP HANA
- 1 additional IP address for takeover
- 1 additional IP address for non-replicated database
- 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 of the replicated SAP HANA database
Instance Number		Number of the SAP HANA database. For system replication also Instance Number + 1 is blocked.
Database user		Database user used by HA/DR provider hook script susCostOpt.py
Database user key		Database user key in <sid> adm`s keystore
SID non-replicated		SAP System Identifier of the non-replicated SAP HANA database
Instance Number		Number of the non-replicated SAP HANA database.
Network mask		
vIP primary		Virtual IP address to be assigned to the primary SAP HANA site
vIP non-replicated		Virtual IP address to be assigned to the non-replicated SAP HANA system (optional)

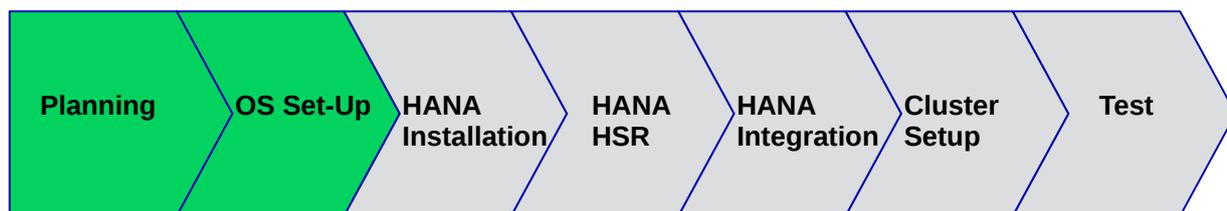
Parameter	Value	Role
Storage		Storage for HDB data and log files is connected “locally” (per node; not shared)
SBD		STONITH device (two for production) or diskless SBD
HAWK Port	<u>7630</u>	
NTP Server		Address or name of your time server

TABLE 2: PARAMETER SHEET WITH VALUES USED IN THIS DOCUMENT

Parameter	Value	Role
Node 1	<u>suse01,</u> <u>192.168.1.11</u>	Cluster node name and IP address.
Node 2	<u>suse02,</u> <u>192.168.1.12</u>	Cluster node name and IP address.
SID	<u>HA1</u>	SAP System Identifier of the replicated SAP HANA database
Instance Number	<u>10</u>	Instance number of the SAP HANA database. For system replication also Instance Number + 1 is blocked.
Database user		Database user used by HA/DR provider hook script susCostOpt.py
Database user key	<u>sus_HA1_costopt</u>	Database user key in <sid> adm`s keystore
SID non-replicated	<u>QAS</u>	SAP System Identifier of the non-replicated SAP HANA database
Instance Number	<u>20</u>	Instance Number of the non-replicated SAP HANA database.

Parameter	Value	Role
Network mask	<u>255.255.255.0</u>	Network mask for SAP HANA's virtual IP address(es)
vIP primary	<u>192.168.1.20</u>	
vIP non-replicated	<u>192.168.1.22</u>	(optional)
Storage		Storage for HDB data and log files is connected "locally" (per node; not shared)
SBD	<u>/dev/disk/by-id/</u> <u>SBDA</u>	STONITH device (two for production) or diskless
HAWK Port	<u>7630</u>	
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-SP3/html/SLES-all/book-sle-deployment.html>). In case of automated installations you can find further information in the *AutoYaST Guide* (<https://documentation.suse.com/sles/15-SP3/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 resource agent for SAP HANA **scale-up** is installed. For the **scale-out** scenario you need a special resource agent. 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

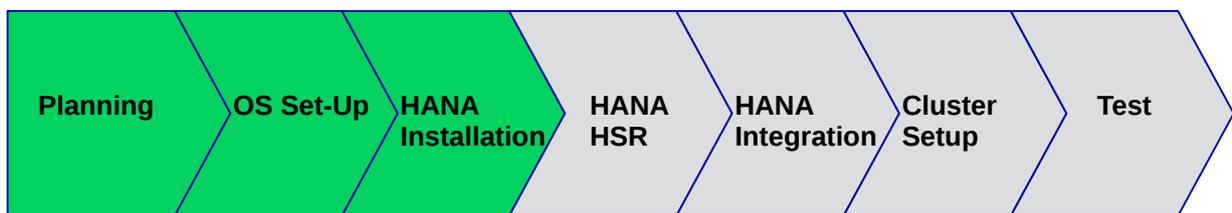
1. Install the High Availability pattern on all nodes

```
# zypper in --type pattern ha_sles
```

2. Install the SAPHanaSR resource agents on all nodes

Optionally, the packages `supportutils-plugin-ha-sap` and `ClusterTools2` can be installed. The first helps collecting data for support requests, the second simplifies common administrative tasks. For more information, see section *Installation and Basic Setup* of the SUSE Linux Enterprise High Availability Extension 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.

PROCEDURE

1. Install the replicating 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 replicating databases are up and running.
4. Install the non-replicating SAP HANA database on the secondary node.
5. Verify that the non-replicating database is up and running.

6.1 Install the replicating 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 Check if the SAP hostagent is installed on all cluster nodes

Check if the SAP hostagent is installed on all cluster nodes. If not, install and enable it now.

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.

6.3 Verify that both databases are up and running

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

```
suse01:haladm> HDB info
USER      PID    PPID  ... COMMAND
haladm    13017  ...  -sh
haladm    13072  ...  \_ /bin/sh /usr/sap/HA1/HDB10/HDB info
haladm    13103  ...  \_ ps fx -U haladm -o
user:8,pid:8,ppid:8,pcpu:5,vsz:10,rss:10,args
haladm    9268  ...  hdbrsutil --start --port 31003 --volume 2 --volumesuffix
mnt00001/hdb00002.00003 --identifier 1580897137
haladm    8911  ...  hdbrsutil --start --port 31001 --volume 1 --volumesuffix
mnt00001/hdb00001 --identifier 1580897100
haladm    8729  ...  sapstart pf=/hana/shared/HA1/profile/HA1_HDB10_suse01
```

```

haladm      8738    ...  \_ /usr/sap/HA1/HDB10/suse02/trace/hdb.sapHA1_HDB10 -d -nw -f /
usr/sap/HA1/HDB10/suse01/daemon.ini pf=/usr/sap/HA1/SYS/profile/HA1_HDB10_suse01
haladm      8756    ...  \_ hdbnameserver
haladm      9031    ...  \_ hdbcompileserver
haladm      9034    ...  \_ hdbpreprocessor
haladm      9081    ...  \_ hdbindexserver -port 31003
haladm      9084    ...  \_ hdbxsengine -port 31007
haladm      9531    ...  \_ hdbwebdispatcher
haladm      8574    ... /usr/sap/HA1/HDB10/exe/sapstartsrv pf=/hana/shared/HA1/profile/
HA1_HDB10_suse01 -D -u haladm

```

6.4 Install the non-replicated SAP HANA on the secondary site

- Stop the secondary (unlimited database)
- Install the non-replicated SAP HANA with memory limits
- Stop the non-replicated SAP HANA

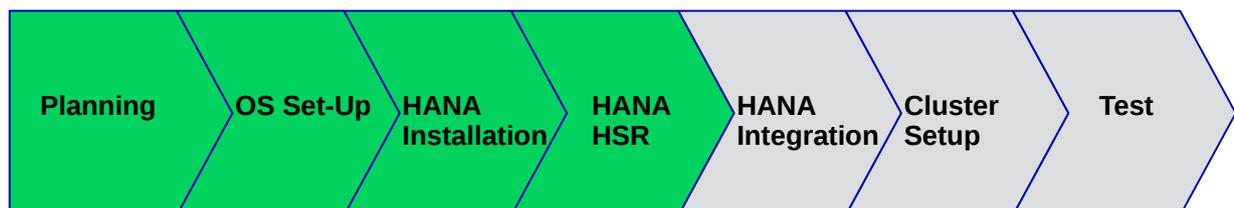
EXAMPLE 2: CHECK THE MEMORY LIMITATIONS FOR THE NON-REPLICATED SAP HANA

```

suse02:~ # su - qasadm
qasadm@suse02:/usr/sap/QAS/HDB20> cd /usr/sap/QAS/SYS/global/hdb/custom/config
qasadm@suse02:/usr/sap/QAS/SYS/global/hdb/custom/config> grep -A1 memorymanager \
global.ini
[memorymanager]
global_allocation_limit = <size_in_mb_for_non_replicated_hana>

```

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 primary database.
3. Register, limit 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 3: SIMPLE BACKUP FOR THE SYSTEM DATABASE AND ALL TENANTS WITH ONE SINGLE BACKUP CALL

As user `<sid>adm` enter the following command:

```
~> hdbsql -u SYSTEM -d SYSTEMDB -i 10 \  
"BACKUP DATA FOR FULL SYSTEM USING FILE ('backup')"
```

You will get a command output similar to the following:

```
0 rows affected (overall time 7255.049 msec; server time 7253.369 msec)
```

EXAMPLE 4: 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 5: ENABLE THE PRIMARY

Enable the primary using the `-sr_enable` option.

```
ha1adm@suse01:/usr/sap/HA1/HDB10> hdbnsutil -sr_enable --name=WDF
nameserver is active, proceeding ...
successfully enabled system as system replication source site
done.
```

EXAMPLE 6: CHECK SR CONFIGURATION ON THE PRIMARY

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

```
ha1adm@suse01:/usr/sap/HA1/HDB10> 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 7: STOP THE SECONDARY

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

```
ha1adm@suse02:/usr/sap/HA1/HDB10> HDB stop
```

EXAMPLE 8: 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-site A>:,}$PWD/data/SSFS_<SID>.DAT
~> rsync -va {<node1-site A>:,}$PWD/key/SSFS_<SID>.KEY
```

EXAMPLE 9: REGISTER THE SECONDARY

The registration of the secondary is triggered by calling `hdbnsutil -sr_register`

```
...
haladm@suse02:/usr/sap/HA1/HDB10> hdbnsutil -sr_register --name=ROT \
  --remoteHost=suse01 --remoteInstance=10 \
  --replicationMode=sync --operationMode=logreplay
adding site ...
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 10: SET MEMORY LIMITS FOR THE SAP HANA SECONDARY

Add the memory limits to the `global.ini`. Keep in mind that SUSE cannot provide a sizing guide here. SAP HANA sizing needs to be done according to respective SAP guidelines.

```
[memorymanager]
global_allocation_limit = <size_in_mb_for_secondary_hana>
```

EXAMPLE 11: SWITCH TABLE PRE-LOAD TO OFF

To allow the SAP HANA secondary with less memory, you need to switch-off table pre-load.

```
[system_replication]
preload_column_tables = false
```

EXAMPLE 12: 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`.

```
haladm@suse02:/usr/sap/HA1/HDB10> HDB start
...
haladm@suse02:/usr/sap/HA1/HDB10> 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 13: CHECKING SYSTEM REPLICATION STATUS DETAILS

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

```
haladm@suse01:/usr/sap/HA1/HDB10> 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 `SAPHanaSR.py`. The hook script `SAPHanaSR.py` is defacto madatory.

Another hook is called by SAP HANA after an SR takeover has happened. This method can be used to remove the SAP HANA memory limit and enable table preload. This HA/DR provider method is `postTakeover()`, the related SUSE hook script is `susCost0pt.py`.

Optionally, 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`. For installation details, refer to manual page `susChkSrv.py(7)`.

Procedure

1. Implement the python hook script SAPHanaSR.py on both sites.
2. Implement the python hook script susCostOpt.py on failover site.
3. Configure the system replication operation mode.
4. Allow <sid>adm to access the cluster on both sites.
5. Create a database user key in <sid>adm's keystore on failover site.
6. Test the hook integration.

This will implement two SAP HANA HA/DR provider hook scripts. The hook script SAPHanaSR.py needs no config parameters. The configuration for susCostOpt.py needs to be adapted to your specific system size and user keystore.



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 SAPHanaSR hook for srConnectionChanged

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

EXAMPLE 14: STOP SAP HANA

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

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

EXAMPLE 15: ADDING SAPHANASR VIA GLOBAL.INI

Use the SAP HANA tools for changing global.ini.

```
[ha_dr_provider_saphanasr]
provider = SAPHanaSR
path = /usr/share/SAPHanaSR/
execution_order = 1

[trace]
ha_dr_saphanasr = info
```

8.2 Implementing susCostOpt hook for postTakeover

Use the hook from the SAPHanaSR package `/usr/share/SAPHanaSR/susCostOpt.py`. The hook must be configured on the takeover node where usually the SAP HANA non-replicated database and the system replication secondary database is running. In `global.ini`, the section `[ha_dr_provider_suscostopt]` needs to be created. The sections `[memorymanager]`, `[system_replication]` and optionally `[trace]` need to be adapted. Refer to manual page `susCostOpt.py(7)` for details on this HA/DR provider hook script, see also `SAPHanaSR-manage-Provider(8)`

EXAMPLE 16: STOP SAP HANA

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

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

EXAMPLE 17: ADDING SUSCOSTOPT VIA GLOBAL.INI

Use the SAP HANA tools for changing `global.ini`.

```
[memorymanager]
global_allocation_limit = <size_in_mb_for_secondary_hana>

[system_replication]
preload_column_tables = false
...

[ha_dr_provider_suscostopt]
provider = susCostOpt
path = /usr/share/SAPHanaSR/
userkey = sus_<SID>_costopt
execution_order = 2

[trace]
ha_dr_suscostopt = info
```

...

8.3 Configuring system replication operation mode

When your system is connected as an SAP HANA SR 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*, not suitable for the cost optimized scenario)

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 "How To Perform System Replication for SAP HANA".

EXAMPLE 18: 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
```

8.4 Allowing <sid>adm to access the cluster

The current version of the SAPHanaSR 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.

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

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.

EXAMPLE 19: ENTRY IN SUDO PERMISSIONS /ETC/SUDOERS.D/SAPHANASR FILE

Basic sudoers entry to allow `<sid> adm` to use the hook SAPHanaSR.

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

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 four 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.

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

8.5 Create a database user key in `<sid> adm`'s keystore

The user keystore needs to be created for the `<sid> adm` of the replicating database. The keystore needs to be created on node 2, where the replication secondary database is running. In this example we use the SAP HANA database user `SYSTEM` for changing the values for memory limit and column pre-load.

EXAMPLE 20: CREATING A DATABASE USER KEY

The example port and password need to be adapted.

```
# su - <sid>adm
~> hdbuserstore SET sus_<SID>_costopt localhost:31013 SYSTEM SuSE1234
~> hdbuserstore LIST sus_<SID>_costopt
```



Important

For security reasons on production systems always a dedicated database user with restricted permission needs to be created. The database user `SYSTEM` must never be used.

Refer to SAP HANA documentation for further information on database users, permissions and the user keystore.

8.6 Start SAP HANA and test the hook integration

After implementing the hooks on both nodes, you should start the productive replicating SAP HANA database on both nodes and check if the hook scripts have been loaded.

Start SAP HANA on both nodes. As user `<sid> adm` either use *HDB* or *sapcontrol*.

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

On both nodes you should check for the SAPHanaSR hook script. Perform the following commands as user `<sid> adm`:

```
<sid>adm@suse02:/usr/sap/<SID>/HDB<instanceNumber>> cdtrace  
<sid>adm@suse02:/usr/sap/<SID>/HDB<instanceNumber>> grep HADR.*load.*SAPHanaS  
nameserver_*.3*.trc
```

On both nodes you might perform the following commands as user root:

```
# grep "sudo.*crm_attribute.*srHook" /var/log/messages
```

On node 2 you should check for the susCostOpt hook script. Perform the following commands as user `<sid> adm`:

```
<sid>adm@suse02:/usr/sap/<SID>/HDB<instanceNumber>> cdtrace  
<sid>adm@suse02:/usr/sap/<SID>/HDB<instanceNumber>> grep HADR.*load.*susCostOpt  
nameserver_*.trc  
<sid>adm@suse02:/usr/sap/<SID>/HDB<instanceNumber>> grep susCostOpt.init  
nameserver_*.3*.trc
```

See also manual pages *SAPHanaSR.py(7)* and *susCostOpt.py(7)*.

9 Configuring the cluster



This chapter describes the configuration of the cluster software SUSE Linux Enterprise High Availability Extension, 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 21: 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 Extension 15 SP3 at <https://documentation.suse.com/sle-ha/15-SP3/single-html/SLE-HA-install-quick/index.html>.

This setup uses unicast (UCAST) for corosync communication (`-u` option). Refer to the <https://documentation.suse.com/sle-ha/15-SP3/single-html/SLE-HA-guide/> 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 "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 (at least one device, in this guide two)
- 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
<code>SBD_WATCHDOG_DEV="/dev/watchdog"</code>	Use watchdog. It is mandatory to use a watchdog. SBD does not work reliable without watchdog. Refer to the SLES manual and SUSE TIDs 7016880 for setting up a watchdog.
<code>SBD_STARTMODE="always"</code>	Start mode. If set to "clean", sbd will only start if the node was previously shut down cleanly or if the slot is empty.
<code>SBD_PACEMAKER="yes"</code>	Check Pacemaker quorum and node health.

```
# egrep -v "(^#|^$)"/etc/sysconfig/sbd
SBD_PACEMAKER="yes"
SBD_STARTMODE="clean"
SBD_DELAY_START="no"
SBD_WATCHDOG_DEV="/dev/watchdog"
SBD_WATCHDOG_TIMEOUT="5"
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"
```

On your specific system, the file might have additional parameters not discussed here.

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                : 1fdc54e8-e2f0-4f02-9e6a-51acbe5656cf
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-575d-4937-a48b-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, read the section *Using SBD as Fencing Mechanism* of the Installation and Setup Quick Start for SUSE Linux Enterprise High Availability Extension 15, and the TIDs 7016880 and 7008216.

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. First, check if all nodes have used the SBD devices. 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
1      suse02      clear
0      suse01      clear
1      suse02      clear
```

You can also call `cs_show_sbd_devices` again.

Next, check if all nodes have joined the cluster. To do so, call `crm_mon`. Use the options `"-r1"` to also see the resources that 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 section *Configuring and Managing Cluster Resources (Command Line)* of the Administration Guide for SUSE Linux Enterprise High Availability Extension 15 SP3 at <https://documentation.suse.com/sle-ha/15-SP3/single-html/SLE-HA-guide/#cha-ha-manual-config>. The manual page `crm(8)` might be useful, too.

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 `msgwait` timeout.

```

suse01:~ # vi crm-bs.txt
# enter the following to crm-bs.txt
property cib-bootstrap-options: \
    have-watchdog="true" \
    stonith-enabled="true" \
    stonith-action="reboot" \
    stonith-timeout="150s"
rsc_defaults rsc-options: \
    resource-stickiness="1000" \
    migration-threshold="3"

```

```
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

For an advanced SBD setup, refer to the SUSE Linux Enterprise High Availability Extension product documentation (for example, visit <https://documentation.suse.com/sle-ha/15-SP3/single-html/SLE-HA-guide/#pro-ha-storage-protect-fencing>). If the preferred node running the primary HANA database always should win in case of split-brain, look up the "Predictable Static Delays" configuration example. See also *Section 13.4.1, "Example for deterministic SBD STONITH"*.

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

9.2.3 Using IPMI as fencing mechanism

This section is only relevant if the recommended disk-based or diskless SBD fencing is not used. For details about IPMI/ILO fencing, read the cluster product documentation (<https://documentation.suse.com/sle-ha/15-SP3/single-html/SLE-HA-guide/>). An example for an IPMI STONITH resource can be found in *Section 13.4.2, "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 Extension product also supports additional fencing mechanism not covered here.

For further information about fencing, read the Administration Guide for SUSE Linux Enterprise High Availability Extension at <https://documentation.suse.com/sle-ha/15-SP3/single-html/SLE-HA-guide/>. 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_SAPHanaTopology_HA1_HDB10 ocf:suse:SAPHanaTopology \
    op monitor interval="10" timeout="600" \
    op start interval="0" timeout="600" \
    op stop interval="0" timeout="300" \
    params SID="HA1" InstanceNumber="10"
clone cln_SAPHanaTopology_HA1_HDB10 rsc_SAPHanaTopology_HA1_HDB10 \
    meta clone-node-max="1" interleave="true"
```

Additional information about all parameters can be found with the command:

```
man ocf_suse_SAPHanaTopology
```

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 SAPHana

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-saphana.txt*, and load it with the following command:

```
crm configure load update crm-saphana.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
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	<p>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.</p> <p>If set to <u>false</u>, 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.</p>
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

Parameter	Description
	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 SAPHana RA can be found with the following command:

```
man ocf_suse_SAPHana
```

```
# vi crm-saphana.txt
# enter the following to crm-saphana.txt
primitive rsc_SAPHana_HA1_HDB10 ocf:suse:SAPHana \
    op start interval="0" timeout="3600" \
    op stop interval="0" timeout="3600" \
    op promote interval="0" timeout="3600" \
    op monitor interval="60" role="Master" timeout="700" \
    op monitor interval="61" role="Slave" timeout="700" \
    params SID="HA1" InstanceNumber="10" PREFER_SITE_TAKEOVER="false" \
        DUPLICATE_PRIMARY_TIMEOUT="7200" AUTOMATED_REGISTER="false" \
    meta priority="100"
ms msl_SAPHana_HA1_HDB10 rsc_SAPHana_HA1_HDB10 \
    meta clone-max="2" clone-node-max="1" interleave="true"
```

Now add the configuration to the cluster.

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

The most important parameters here are again SID and InstanceNumber. Beside these parameters, typical tuneables are the timeout values for the operations (start, promote, monitors, stop). The parameter *AUTOMATED_REGISTER* can be used to adapt the level of system replication automation. The general resource meta attribute *priority* can be used together with the optional priority fencing to make an HANA primary node surviving in case of split-brain. See the SUSE Linux Enterprise High Availability Extension product documentation for details (<https://documentation.suse.com/sle-ha/15-SP3/single-html/SLE-HA-guide/#pro-ha-storage-protect-fencing>).

9.2.7 Adding a virtual IP address for the primary site

The last resource to be added for SAPHanaSR is covering the virtual IP address.

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

```
primitive rsc_ip_HA1_HDB10 ocf:heartbeat:IPaddr2 \  
  op monitor interval="10s" timeout="20s" \  
  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.

9.2.8 Constraints for SAPHanaSR

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

```
# vi crm-cs.txt  
# enter the following to crm-cs.txt  
colocation col_saphana_ip_HA1_HDB10 3000: rsc_ip_HA1_HDB10:Started \  
  msl_SAPHana_HA1_HDB10:Master  
order ord_SAPHana_HA1_HDB10 Optional: cln_SAPHanaTopology_HA1_HDB10 \  
  msl_SAPHana_HA1_HDB10
```

Load the file to the cluster.

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

9.2.9 Adding the cluster resource for the non-replicated SAP HANA database

For the non-replicated SAP HANA database, a new resource is added to the cluster. In previous versions of this document we used the resource agent SAPDatabase to control that database. The new architecture now uses SAPIInstance to start, stop and monitor this cluster component. The reason for that change is that SAPDatabase is using the SAP host agent API. The SAP host agent itself needs user secure keys to communicate with the cluster. This configuration is too complex and error-prone. SAPIInstance uses the `sapstartsrv` API to do the work. This should solve the issue. The new concept has already been published in a SUSE #towardsZeroDowntime blog at <https://suse.com/c/sap-hana-cost-optimized-an-alternative-route-is-available/> 

```
# vi crm-si.txt  
# enter the following to crm-si.txt  
primitive rsc_SAP_QAS_HDB20 ocf:heartbeat:SAPIInstance \  
  op monitor interval="10s" timeout="20s" \  
  params ip="192.168.1.20"
```

```

params InstanceName="QAS_HDB20_suse02" \
        MONITOR_SERVICES="hbindexserver|hdbnameserver" \
        START_PROFILE="/usr/sap/QAS/SYS/profile/QAS_HDB20_suse02" \
op start interval="0" timeout="600" \
op monitor interval="120" timeout="700" \
op stop interval="0" timeout="300"

```

Load the resource definition into the cluster:

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

9.2.10 Adding cluster rules for automatic shutdown of the non-replicated SAP HANA

In the following example, again suse01 and suse02 are used as the two active cluster nodes.

```

# vi crm-con.txt
# enter the following to crm-con.txt
location loc_QAS_never_on_suse01 rsc_SAP_QAS_HDB20 -inf: suse01
colocation col_QAS_never_with_HA1ip -inf: rsc_SAP_QAS_HDB20:Started \
        rsc_ip_HA1_HDB10
order ord_QASstop_before_HA1-promote mandatory: rsc_SAP_QAS_HDB20:stop \
        msl_SAPHana_HA1_HDB10:promote

```

Load the resource definition into the cluster

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

10 Testing the cluster



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

For any cluster setup testing is crucial. Make sure that all test cases derived from your organizations or from customer expectations are fully implemented and successfully passed. Otherwise the project is likely to fail in production.

If not described differently, the test prerequisite is always that both nodes are booted, normal members of the cluster and that the HANA RDBMS is running. There are no left-over migration constraints or resource failures contained in the cluster information base (CIB). The system replication is in sync (*SOK*) and the cluster is not performing any action, this means it is in the state *S_IDLE*.

This can be checked, for example, with the following command sequence:

```
# crm_mon -lr
# crm configure show | grep cli-
# SAPHanaSR-showAttr
# cs_clusterstate -i
```

See also the manual pages [SAPHanaSR-showAttr\(8\)](#), [crm_mon\(8\)](#), [crm\(8\)](#), [cs_clusterstate\(8\)](#), [SAPHanaSR_maintenance_examples\(7\)](#).

10.1 Test cases for semi-automation

For the following test descriptions we assume these following parameter values: * [PRE-FER_SITE_TAKEOVER="false"](#) * [AUTOMATED_REGISTER="false"](#).



Note

The following tests are designed to run in a sequence. Each test depends on the exit state of the preceding tests.

10.1.1 Tests for primary database or node

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

EXAMPLE 22: TEST_STOP_PRIMARY_DB_SITE_A_SEMI

COMPONENT:

- Primary Database

DESCRIPTION:

- Stop Primary on site A (node 1)

TEST PROCEDURE:

1. Stop the SAP HANA database as user <sid> adm

```
suse01:~ # su - haladm
haladm@suse01:/usr/sap/HA1/HDB10> HDB stop
```

EXPECTED:

- Primary restarts on site A (PREFER_SITE_TAKEOVER = false) until failcount > = migration-threshold
- If takeover occurs:
 - non-replicated database is stopped on node 2 (site B)
 - Secondary database is promoted as primary

RECOVERY PROCEDURE:

- No recovery needed, if no takeover did occur
- Recovery after takeover:
 1. Register site A to site B
 2. Resource cleanup for site A

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

EXAMPLE 23: TEST_STOP_PRIMARY_DB_SITE_B_SEMI

COMPONENT:

- Primary Database

DESCRIPTION:

- Stop Primary on site B (node 2)

TEST PROCEDURE:

1. Stop the SAP HANA database as user <sid> adm

```
haladm@suse02:/usr/sap/HA1/HDB10> HDB stop
```

EXPECTED:

- Primary restarts on site B (PREFER_SITE_TAKEOVER = false) until failcount > = migration-threshold
- Non-replicated database still stopped on node 2 (site B)
- If takeover occurs:
 - Secondary database is promoted as primary
 - non-replicated database is started on node 2 (site B)

RECOVERY PROCEDURE:

- No recovery needed if no takeover did occur
- Recovery after takeover:
 1. Register site B to site A
 2. Resource cleanup for site B

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

EXAMPLE 24: TEST CRASH_PRIMARY_DB_SITE_A_SEMI

COMPONENT:

- Primary Database

DESCRIPTION:

- Kill Primary on site A (node 1)

TEST PROCEDURE:

1. Kill (send signal to) the SAP HANA database as user <sid> adm

```
haladm@suse01:/usr/sap/HA1/HDB10> HDB kill-9
```

EXPECTED:

- Primary restarts on site A (PREFER_SITE_TAKEOVER = false) until failcount > = migration-threshold
- If takeover occurs:
 - Non-replicated database is stopped on node 2 (site B)
 - Secondary database is promoted as primary

RECOVERY PROCEDURE:

- No recovery needed if no takeover did occur
- Recovery after takeover:
 1. Register site A to site B
 2. Resource cleanup for site A

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

EXAMPLE 25: TEST CRASH_PRIMARY_DB_SITE_B_SEMI

COMPONENT:

- Primary Database

DESCRIPTION:

- Kill Primary on site B (node 2)

TEST PROCEDURE:

1. Kill Primary on site B (node 2) as user <sid> adm

```
ha1adm@suse02:/usr/sap/HA1/HDB10> HDB kill-9
```

EXPECTED:

- Primary restarts on site B (PREFER_SITE_TAKEOVER = false) until failcount > = migration-threshold
- Non-replicated database still stopped on node 2 (site B)

RECOVERY PROCEDURE:

- No recovery needed if no takeover did occur
- Recovery after takeover:
 1. Register site B to site A
 2. Resource cleanup for site B

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

EXAMPLE 26: TEST_CRASH_PRIMARY_NODE_SITE_A_SEMI

COMPONENT:

- Cluster Node

DESCRIPTION:

- Crash node 1 (site A)

TEST PROCEDURE:

1. Crash the node by proc-sysrq-trigger as user root

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

EXPECTED:

- Non-replicated SAP HANA stopped on node 2
- Cluster takeover to site B
- Non-replicated database is stopped on node 2 (site B)
- Secondary database is promoted as primary

RECOVERY PROCEDURE:

- Recovery after takeover:
 1. Optionally clean up sbd slot for node 1
 2. Register site A to site B
 3. Start cluster framework on node 1
 4. Wait until node 1 joins the cluster

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

EXAMPLE 27: TEST_CRASH_PRIMARY_NODE_SITE_B_SEMI

COMPONENT:

- Cluster Node

DESCRIPTION:

- Crash node 2 (site B)

TEST PROCEDURE:

1. Crash the node by proc-sysrq-trigger as user root

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

EXPECTED:

- Cluster takeover to site A
- Non-replicated database not available (no takeover to site A)

RECOVERY PROCEDURE:

- Recovery after takeover:
 1. Optionally clean up sbd slot for node 2
 2. Register site B to site A
 3. Start cluster framework on node 2
 4. Wait until node 2 joins the cluster

10.1.2 Tests for secondary database or node

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

EXAMPLE 28: TEST_STOP_SECONDARY_DB_SITE_B_SEMI

COMPONENT:

- Secondary Database

DESCRIPTION:

- Stop secondary database on node 2 (site B)

TEST PROCEDURE:

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

```
ha1adm@suse02:/usr/sap/HA1/HDB10> HDB stop
```

EXPECTED:

- Cluster restarts Secondary on node 2 (site B)
- non-replicated database not affected on node 2 (site B)

RECOVERY PROCEDURE:

1. Wait and see
2. Resource cleanup for site B

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

EXAMPLE 29: TEST CRASH_SECONDARY_DB_SITE_B_SEMI

COMPONENT:

- Secondary Database

DESCRIPTION:

- Crash secondary database on node 2 (site B)

TEST PROCEDURE:

1. Kill (send signal to) the secondary SAP HANA database as user <sid> adm

```
ha1adm@suse02:/usr/sap/HA1/HDB10> HDB kill-9
```

EXPECTED:

- Cluster restarts Secondary on node 2 (site B)
- Non-replicated database not affected on node 2 (site B)

RECOVERY PROCEDURE:

1. Wait and see
2. Resource cleanup for site B

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

EXAMPLE 30: TEST CRASH_SECONDARY_NODE_SITE_B_SEMI

COMPONENT:

- Cluster Node

DESCRIPTION:

- Crash node 2 (site B)

TEST PROCEDURE:

1. Crash the node by proc-sysrq-trigger as user root

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

EXPECTED:

- No takeover of node 2 resources to site A
- Non-replicated database not available (no takeover to site A)

RECOVERY PROCEDURE:

- Recovery after node 2 is back:
 1. Optionally clean up sbd slot for node 2
 2. Start cluster framework on node 2
 3. Wait until node 2 joins the cluster

10.1.3 Tests for non-replicated database

10.1.3.1 Test: Stop non-replicated database on site B (node 2)

EXAMPLE 31: TEST STOP_NONSR_DB_SITE_B_SEMI

COMPONENT:

- Non-Replicated Database

DESCRIPTION:

- Stop non-replicated database node 2 (site B)

TEST PROCEDURE:

1. Kill (send signal to) the non-replicated SAP HANA database as user <sid> adm

```
qasadm@suse02:/usr/sap/QAS/HDB20> HDB stop
```

EXPECTED:

- Cluster restarts non-replicated database on node 2 (site B)
- Secondary database is not affected

RECOVERY PROCEDURE:

1. Clean up non-replicated database resource

10.1.3.2 Test: Crash non-replicated database on site B (node 2)

EXAMPLE 32: TEST CRASH_NONSR_DB_SITE_B_SEMI

COMPONENT:

- Non-Replicated Database

DESCRIPTION:

- Crash non-replicated database on node 2 (site B)

TEST PROCEDURE:

1. Kill (send signal to) the non-replicated SAP HANA database as user <sid> adm

```
qasadm@suse02:/usr/sap/QAS/HDB20> HDB kill-9
```

EXPECTED:

- Cluster restarts non-replicated database on node 2 (site B)
- Secondary database is not affected

RECOVERY PROCEDURE:

1. Clean up non-replicated database resource

10.1.4 Tests for other components

10.1.4.1 Test: Failure of dedicated replication LAN

EXAMPLE 33: TEST FAIL_NETWORK_SR_SEMI

COMPONENT:

- Replication Network

DESCRIPTION:

- Pull LAN port down or block network packets for system replication. Corosync network still available.

EXPECTED:

- System replication status fall down to status *SFAIL*
- Primary stays on node 1 (site A)
- No cluster takeover
- Non-replicated database not affected on node 2 (site B)

RECOVERY PROCEDURE:

1. Re-establish network connection
2. Wait until System replication status is *SOK* again

10.1.5 Test maintenance procedures

Also test the maintenance procedures mentioned in section [Section 11.3, "Maintenance"](#) (page 77).

10.2 Test cases for full automation

For the following test descriptions, we assume the following parameter values: * PRE-FER_SITE_TAKEOVER="false" * AUTOMATED_REGISTER="true".



Note

The following tests are designed to run in a sequence. Each test depends on the exit state of the preceding tests.

10.2.1 Tests for primary database or node

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

EXAMPLE 34: TEST STOP_PRIMARY_DB_SITE_A_FULL

COMPONENT:

- Primary Database

DESCRIPTION:

- Stop primary database on site A (node 1)

TEST PROCEDURE:

1. Stop the SAP HANA database as user <sid> adm

```
haladm@suse01:/usr/sap/HA1/HDB10> HDB stop
```

EXPECTED:

- Primary restarts on site A (PREFER_SITE_TAKEOVER = false) until failcount > = migration-threshold
- If takeover occurs:
 - Non-replicated database is stopped on node 2 (site B)
 - Secondary database is promoted as primary
 - SiteA is automatically registered to SiteB

RECOVERY PROCEDURE:

- No recovery needed, if no takeover did occur
- Recovery after takeover:
 1. Resource cleanup for site A

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

EXAMPLE 35: TEST_STOP_PRIMARY_DB_SITE_B_FULL

COMPONENT:

- Primary Database

DESCRIPTION:

- Stop primary database on site B (node 2)

TEST PROCEDURE:

1. Stop the SAP HANA database as user <sid> adm

```
ha1adm@suse02:/usr/sap/HA1/HDB10> HDB stop
```

EXPECTED:

- Primary restarts on site B (PREFER_SITE_TAKEOVER = false) until failcount > = migration-threshold
- Non-replicated database still stopped on node 2 (site B)
- If takeover occurs:
 - Secondary database is promoted as primary
 - non-replicated database is started on node 2 (site B)
 - SiteB is automatically registered to SiteA

RECOVERY PROCEDURE:

- No recovery needed if no takeover did occur
- Recovery after takeover:
 1. Resource cleanup for site B

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

EXAMPLE 36: TEST CRASH_PRIMARY_DB_SITE_A_FULL

COMPONENT:

- Primary Database

DESCRIPTION:

- Kill primary database on site A (node 1)

TEST PROCEDURE:

1. Kill (send signal to) the SAP HANA database as user <sid> adm

```
haladm@suse01:/usr/sap/HA1/HDB10> HDB kill-9
```

EXPECTED:

- Primary restarts on site A (PREFER_SITE_TAKEOVER = false) until failcount > = migration-threshold
- If takeover occurs:
 - Non-replicated database is stopped on node 2 (site B)
 - Secondary database is promoted as primary
 - SiteA is registered to SiteB

RECOVERY PROCEDURE:

- No recovery needed if no takeover did occur
- Recovery after takeover:
 1. Resource cleanup for site A

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

EXAMPLE 37: TEST CRASH_PRIMARY_DB_SITE_B_FULL

COMPONENT:

- Primary Database

DESCRIPTION:

- Kill primary database on site B (node 2)

TEST PROCEDURE:

1. Kill primary database on site B (node 2) as user <sid> adm

```
haladm@suse02:/usr/sap/HA1/HDB10> HDB kill-9
```

EXPECTED:

- Primary restarts on site B (PREFER_SITE_TAKEOVER = false) until failcount > = migration-threshold
- Non-replicated database still stopped on node 2 (site B)
- If takeover occurs:
 - Secondary database is promoted as primary
 - Non-replicated database is started on node 2 (site B)
 - SiteB is automatically registered to SiteA

RECOVERY PROCEDURE:

- No recovery needed if no takeover did occur
- Recovery after takeover:
 1. Resource cleanup for site B

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

EXAMPLE 38: TEST CRASH_PRIMARY_NODE_SITE_A_FULL

COMPONENT:

- Cluster Node

DESCRIPTION:

- Crash node 1 (site A)

TEST PROCEDURE:

1. Crash the node by proc-sysrq-trigger as user root

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

EXPECTED:

- Non-replicated SAP HANA stopped on node 2
- Cluster takeover to site B
- Non-replicated database is stopped on node 2 (site B)
- Secondary database is promoted as primary
- Later, when node 1 joins the cluster again:
 - SiteA is registered to SiteB

RECOVERY PROCEDURE:

- Recovery after takeover:
 1. Optionally clean up sbd slot for node 1
 2. Start cluster framework on node 1
 3. Wait until node 1 joins the cluster

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

EXAMPLE 39: TEST CRASH_PRIMARY_NODE_SITE_B_FULL

COMPONENT:

- Cluster Node

DESCRIPTION:

- Crash node 2 (site B) as user root

TEST PROCEDURE:

1. Crash the node by proc-sysrq-trigger

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

EXPECTED:

- Cluster takeover to site A
- Non-replicated database not available (no takeover to site A)
- Later, when node 2 joins the cluster again:
 - SiteB is registered to SiteA
 - Non-replicated database will be started

RECOVERY PROCEDURE:

- Recovery after takeover:
 1. Optionally clean up sbd slot for node 2
 2. Start cluster framework on node 2
 3. Wait until node 2 joins the cluster

10.2.2 Tests for secondary database or node

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

EXAMPLE 40: TEST_STOP_SECONDARY_DB_SITE_B_FULL

COMPONENT:

- Secondary Database

DESCRIPTION:

- Stop secondary database on node 2 (site B)

TEST PROCEDURE:

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

```
haladm@suse02:/usr/sap/HA1/HDB10> HDB stop
```

EXPECTED:

- Cluster restarts Secondary on node 2 (site B)
- Non-replicated database not affected on node 2 (site B)

RECOVERY PROCEDURE:

1. Wait and see
2. Resource cleanup for site B

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

EXAMPLE 41: TEST_CRASH_SECONDARY_DB_SITE_B_FULL

COMPONENT:

- Secondary Database

DESCRIPTION:

- Crash secondary database on node 2 (site B)

TEST PROCEDURE:

1. Kill (send signal to) the secondary SAP HANA database as user <sid> adm

```
haladm@suse02:/usr/sap/HA1/HDB10> HDB kill-9
```

EXPECTED:

- Cluster restarts Secondary on node 2 (site B)
- Non-replicated database not affected on node 2 (site B)

RECOVERY PROCEDURE:

1. Wait and see
2. Resource cleanup for site B

10.2.2.3 Test: Crash the secondary node on Site B (node 2)

EXAMPLE 42: TEST_CRASH_SECONDARY_NODE_SITE_B_FULL

COMPONENT:

- Cluster Node

DESCRIPTION:

- Crash node 2 (site B)

TEST PROCEDURE:

1. Crash the node by proc-sysrq-trigger as user root

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

EXPECTED:

- No takeover of node 2 resources to site A
- Non-replicated database not available (no takeover to site A)

RECOVERY PROCEDURE:

- Recovery after node 2 is back:
 1. Optionally clean up sbd slot for node 2
 2. Start cluster framework on node 2
 3. Wait until node 2 joins the cluster

10.2.3 Tests for non-replicated database

10.2.3.1 Test: Stop non-replicated database on site B (node 2)

EXAMPLE 43: TEST STOP_NONSR_DB_SITE_B_FULL

COMPONENT:

- Non-Replicated Database

DESCRIPTION:

- Stop non-replicated database node 2 (site B)

TEST PROCEDURE:

1. Kill (send signal to) the non-replicated SAP HANA database as user <sid> adm

```
qasadm@suse02:/usr/sap/QAS/HDB20> HDB stop
```

EXPECTED:

- Cluster restarts non-replicated database on node 2 (site B)
- Secondary database is not affected

RECOVERY PROCEDURE:

1. Clean up non-replicated database resource

10.2.3.2 Test: Crash non-replicated database on site B (node 2)

EXAMPLE 44: TEST CRASH_NONSR_DB_SITE_B_FULL

COMPONENT:

- Non-Replicated Database

DESCRIPTION:

- Crash non-replicated database on node 2 (site B)

TEST PROCEDURE:

1. Kill (send signal to) the non-replicated SAP HANA database as user <sid> adm

```
qasadm@suse02:/usr/sap/QAS/HDB20> HDB kill-9
```

EXPECTED:

- Cluster restarts non-replicated database on node 2 (site B)
- Secondary database is not affected

RECOVERY PROCEDURE:

1. Clean up non-replicated database resource

10.2.4 Tests for other components

10.2.4.1 Test: Failure of dedicated replication LAN

EXAMPLE 45: TEST FAIL_NETWORK_SR_FULL

COMPONENT:

- Replication Network

DESCRIPTION:

- Pull LAN port down or block network packets for system replication, Corosync network still available.

EXPECTED:

- System replication status fall down to status *SFAIL*
- Primary stays on node 1 (site A)
- No cluster takeover
- Non-replicated database not affected on node 2 (site B)

RECOVERY PROCEDURE:

1. Re-establish network connection
2. Wait until system replication status is *SOK* again

10.2.5 Test maintenance procedures

Also, test the maintenance procedures mentioned in section *Section 11.3, "Maintenance"* (page 77).

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="false"`, `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.

! Important

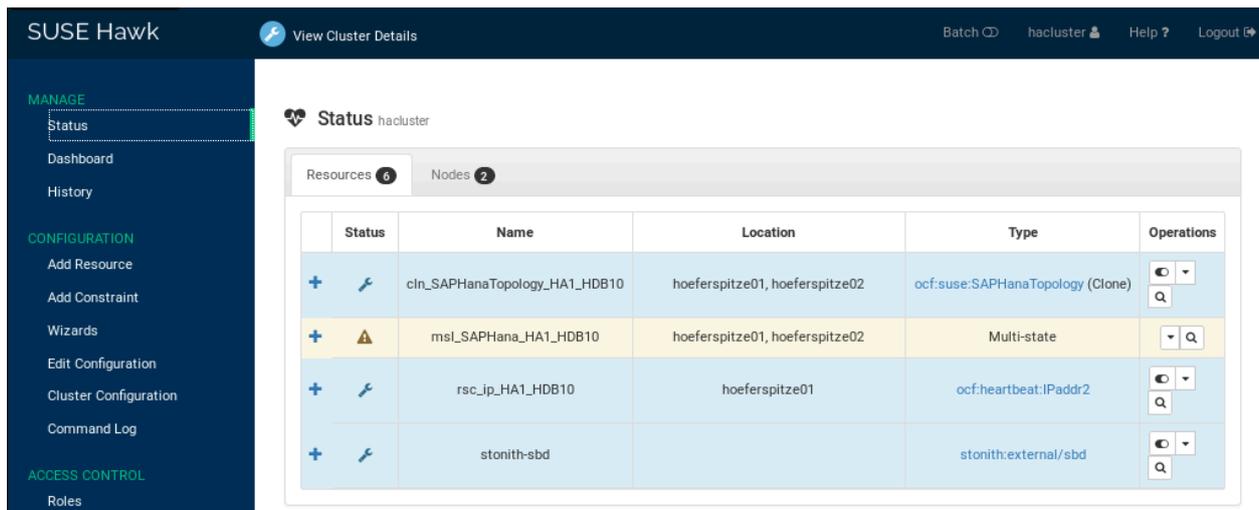
As "migrating" or "moving" resources in crm-shell, HAWK or other tools would add client-prefer location rules, support is limited to maintenance procedures described in this document. See ??? and [Section 11.3, "Maintenance" \(page 77\)](#) for proven procedures.

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.



The screenshot shows the SUSE Hawk web interface for monitoring a cluster. The top navigation bar includes 'View Cluster Details', 'Batch', 'hacluster', 'Help?', and 'Logout'. The left sidebar has sections for 'MANAGE' (Status, Dashboard, History) and 'CONFIGURATION' (Add Resource, Add Constraint, Wizards, Edit Configuration, Cluster Configuration, Command Log). The main content area displays the 'Status hacluster' page with 'Resources 6' and 'Nodes 2'. A table lists the resources:

Status	Name	Location	Type	Operations
+	cln_SAPHanaTopology_HA1_HDB10	hoeferspitze01, hoeferspitze02	ocf:suse:SAPHanaTopology (Clone)	
+	msI_SAPHana_HA1_HDB10	hoeferspitze01, hoeferspitze02	Multi-state	
+	rsc_ip_HA1_HDB10	hoeferspitze01	ocf:heartbeat:IPAddr2	
+	stonith-sbd		stonith:external/sbd	

FIGURE 10: 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 cost 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 77).

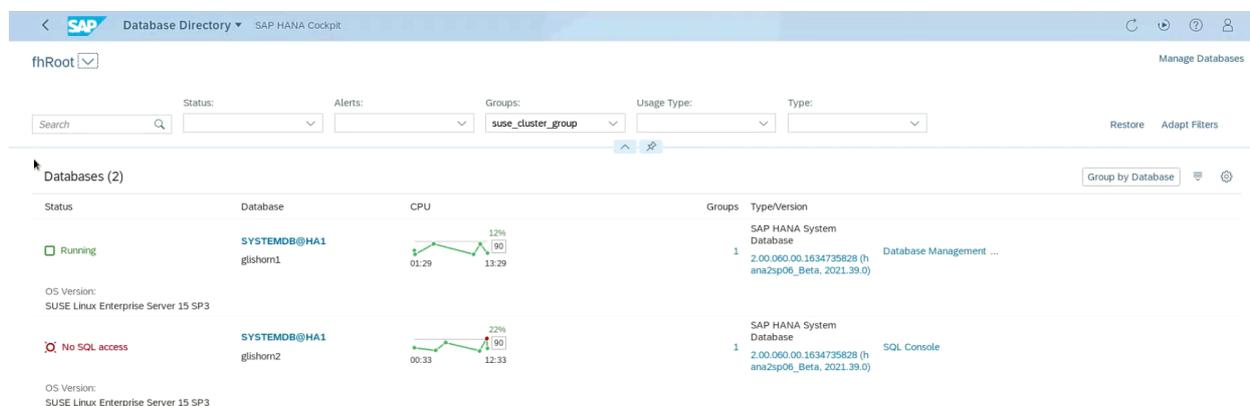


FIGURE 11: SAP HANA COCKPIT – DATABASE DIRECTORY

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 `-l` tells `crm_mon` to output the status once instead of periodically.

```
Cluster Summary:
* Stack: corosync
* Current DC: suse01 (version 2.0.4+20200616.2deceaa3a-3.6.1-2.0.4+20200616.2deceaa3a)
- partition with quorum
* Last updated: Mon Jun 21 19:03:18 2021
* Last change: Mon Jun 21 19:02:41 2021 by root via crm_attribute on suse01
* 2 nodes configured
* 7 resource instances configured

Node List:
* Online: [ suse01 suse02 ]

Full List of Resources:
* rsc_SAP_QAS_HDB20 (ocf::heartbeat:SAPInstance): Started suse02
* stonith-sbd (stonith:external/sbd): Started suse01
* Clone Set: cln_SAPHanaTopology_HA1_HDB10 [rsc_SAPHanaTopology_HA1_HDB10]:
  * Started: [ suse01 suse02 ]
* Clone Set: msl_SAPHana_HA1_HDB10 [rsc_SAPHana_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. If you have installed the ClusterTools2 package, also have a look at manual pages `cs_clusterstate(8)` and `cs_show_hana_info(8)`.

11.2.4 SAPHanaSR command line tools

To show some SAPHana or SAPHanaTopology resource agent internal values, call the program `SAPHanaSR-showAttr`. The internal values, the storage location and their parameter names may change in a next version of this document. The command `SAPHanaSR-showAttr` will always fetch the values from the correct storage location.

Do not use cluster commands like `crm_attribute` to fetch the values directly from the cluster. If you use such commands, your methods will be broken when you need to move an attribute to a different storage place or even out of the cluster. At first `SAPHanaSR-showAttr` is a test program only and should not be used for automated system monitoring.

```
suse01:~ # SAPHanaSR-showAttr
Host \ Attr clone_state remoteHost roles      ... site  srmode sync_state ...
-----
suse01    PROMOTED    suse02    4:P:master1:... WDF    sync  PRIM    ...
suse02    DEMOTED    suse01    4:S:master1:... ROT    sync  SOK    ...
```

`SAPHanaSR-showAttr` also supports other output formats such as **script**. The script format is intended to allow running filters. The SAPHanaSR package beginning with version 0.153 additionally provides a filter engine `SAPHanaSR-filter`. Combining `SAPHanaSR-showAttr` with output format `script` and `SAPHanaSR-filter`, you can define effective queries:

```
suse01:~ # SAPHanaSR-showAttr --format=script | \
  SAPHanaSR-filter --search='remote'
Mon Jun 21 19:55:45 2021; Hosts/suse01/remoteHost=suse02
Mon Jun 21 19:55:45 2021; Hosts/suse02/remoteHost=suse01
```

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

In our example, the administrator killed the primary SAP HANA instance using the command `HDB kill-9`. This happened around 8:10 pm.

```
suse01:~ # hb_report -f 19:00
```

```

INFO: suse01# The report is saved in ./hb_report-Mon-21-Jun-2021.tar.gz
INFO: suse01# Report timespan: 06/21/21 19:00:00 - 06/21/21 20:26:33
INFO: suse01# Thank you for taking time to create this report.
suse01:~ # SAPHanaSR-replay-archive --format=script \
./hb_report-Mon-21-Jun-2021.tar.gz | \
SAPHanaSR-filter --search='roles' --filterDouble
Mon Jun 21 19:38:01 2021; Hosts/suse01/roles=4:P:master1:master:worker:master
Mon Jun 21 19:38:01 2021; Hosts/suse02/roles=4:S:master1:master:worker:master
Mon Jun 21 20:11:37 2021; Hosts/suse01/roles=1:P:master1::worker:
Mon Jun 21 20:15:43 2021; Hosts/suse02/roles=4:P: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 20:11 (CET) suddenly the primary on suse01 died - it was falling down to 1:P.

The cluster did jump in and initiated a takeover. At 20:15 (CET) the former secondary was detected as new running master (changing from 4:S to 4:P). See manual pages SAPHanaSR-showAttr(8), SAPHanaSR-replay-archive(8) and crm_report(8) for more information.

11.2.5 SAP HANA LandscapeHostConfiguration

To check the status of an SAP HANA database and to find out if the cluster should react, use the script **landscapeHostConfiguration** 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 SAPHana 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.

Return Code	Interpretation
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 the SUSE Linux Enterprise High Availability Extension, 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-SP3/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 Extension Administration Guide, chapter *Upgrading Your Cluster and Updating Software Packages* at <https://documentation.suse.com/sle-ha/15-SP3/single-html/SLE-HA-guide/#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 46: 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 msl_SAPHana_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
```

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 HANA commands.

Pre-Migration Tasks

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

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

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 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-name>
```

- 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-name> off
```

The following paragraphs explain how to use the cluster to partially automate the migration. For the described attribute query using `SAPHanaSR-showAttr` and `SAPHanaSR-filter` you need at least `SAPHanaSR` with package version 0.153.

EXAMPLE 48: 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-name> 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.

Note

The `crm resource` command **move** was previously named **migrate**. The **migrate** command is still valid but already known as obsolete.

- 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".

```
suse01:~ # SAPHanaSR-showAttr --format=script | \  
SAPHanaSR-filter --search='roles'  
Mon Jun 21 19:38:50 2021; Hosts/suse01/roles=1:P:master1::worker:  
Mon Jun 21 19:38:50 2021; Hosts/suse02/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-name>
```



Note

The `crm resource clear` command was previously named **unmigrate**. The **unmigrate** command is still valid but already known as obsolete.

- 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".

```
suse01:~ # SAPHanaSR-showAttr --format=script | \
SAPHanaSR-filter --search='roles'
Mon Jun 21 19:38:50 2021; Hosts/suse01/roles=4:S:master1::worker:
Mon Jun 21 19:38:50 2021; Hosts/suse02/roles=4:P:master1:master:worker:master
```

- You should revert the SAP HANA roles back soon, to get the non-replicated database also up and running again.

11.3.4 Reverting to original SAP HANA roles after takeover to secondary site

In the following procedure the HANA primary role is moved back to node1. node2 is registered as the new secondary and the non-replicated HANA is started on node2. The site name to be registered is WDF. Make sure this is identical to the one used before.

EXAMPLE 49: REVERT SAP HANA ROLES BACK AFTER FAILURE ON SUSE01

- Register the former primary as new secondary as user `<sid> adm`.

```
haladm@suse01:/usr/sap/HA1/HDB10> hdbnsutil -sr_register --name=WDF \
--remoteHost=suse02 --remoteInstance=10 \
```

```
--replicationMode=sync --operationMode=logreplay
```

- Clean up the failcount for SAP HANA resource as user root.

```
# crm configure show rsc_SAPHana_HA1_HDB10 | grep AUTOMATED_REGISTER  
# crm resource cleanup msl_SAPHana_HA1_HDB10 suse01
```

- Recover the SAP HANA global.ini back to initial state, as user *<sid> adm*.

```
haladm@suse02:/usr/sap/HA1/HDB10> cdco  
haladm@suse02:/usr/sap/HA1/HDB10> cp global.ini global.ini.bak  
haladm@suse02:/usr/sap/HA1/HDB10> vi global.ini  
[memorymanager]  
global_allocation_limit = <size_in_mb_for_secondary_hana>  
...  
[system_replication]  
preload_column_tables = false
```

- Move the SAP HANA Primary back to suse01, as root user.

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

- Wait until the cluster has finished the transition and is idle. Then remove the migration constraint from CIB.

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

11.4 Support

There are two channels available for opening support requests. For issues which might also need SAP to investigate, the preferred method is to open an SAP ticket on support queue BC-OP-LNX-SUSE. See SAP note 1056161; find the link in *???TITLE???*.

The other channel is to use the SUSE support only. SUSE customer center (SCC) is the central access point for managing support entitlements and for opening support requests. It is available at <https://scc.suse.com/login>.

More information on how to access support can be found at <https://www.suse.com/support/> and <https://www.suse.com/support/faq/>.

The SUSE Linux Enterprise Server for SAP Applications product documentation explains how to collect information usually needed during a support request: <https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-adm-support.html>.

See also manual pages `crm_report(8)`, `supportconf(8)`, `cs_show_hana_info(8)`, `ha_related_suse_tids(7)`.

In addition, there are SUSE support Technical Information Documents (TIDs) available, for example:

- Diagnostic Data Collection Master TID (7024037)
- Indepth HANA Cluster Debug Data Collection (PACEMAKER, SAP) (7022702)
- SLES for SAP - How To Engage SAP and SUSE to address Product Issues (7021182)

The SUSE support knowledgebase containing the TIDs is available at <https://www.suse.com/support/kb/>.

12 References

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

12.1 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 Extension

<https://documentation.suse.com/sle-ha/15-SP3/single-html/SLE-HA-guide/>

Deployment guide for SUSE Linux Enterprise Server

<https://documentation.suse.com/sles/15-SP3/html/SLES-all/book-sle-deployment.html>

Tuning guide for SUSE Linux Enterprise Server

<https://documentation.suse.com/sles/15-SP3/html/SLES-all/book-sle-tuning.html>

Storage administration guide for SUSE Linux Enterprise Server

<https://documentation.suse.com/sles/15-SP3/single-html/SLES-storage/>

SUSE Linux Enterprise Server Persistent Memory Guide

<https://documentation.suse.com/sles/15-SP3/html/SLES-all/cha-nvdim.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/communities/conversations/xfsthefile-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-SP3/html/SLES-all/book-rmt.html>

SUSE Customer Center Frequently Asked Questions

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

12.2 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_SAPHana(7)
- ocf_suse_SAPHanaController(7)
- ocf_suse_SAPHanaTopology(7)
- SAHanaSR(7)
- SAPHanaSR-ScaleOut(7)
- SAPHanaSR-filter(8)
- 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)

- SAPHanaSR.py(7)
- SAPHanaSrMultiTarget.py(7)
- saptune(8)
- sbd(8)
- stonith_sbd(7)
- sudo(8)
- sudoers(5)
- supportconfig(8)
- susChkSrv.py(7)
- susCostOpt.py(7)
- susTkOver.py(7)
- systemctl(8)
- systemd-cgls(8)
- votequorum(5)
- zypper(8)

12.3 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 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>

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> ↗

12.4 Related SUSE blogs

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> 

12.5 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> 

12.6 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 Revision 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> 

12.7 Pacemaker

Pacemaker Project Documentation

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

13 Examples

13.1 Example ha-cluster-init configuration

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

Add another heartbeat line (y/n)? y
Address for ring1 [192.168.2.11]
Port for ring1 [5407]
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'
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

Configure Qdevice/Qnetd:
QDevice participates in quorum decisions. With the assistance of
a third-party arbitrator Qnetd, it provides votes so that a cluster
is able to sustain more node failures than standard quorum rules
allow. It is recommended for clusters with an even number of nodes
and highly recommended for 2 node clusters.
```

```
Do you want to configure QDevice (y/n)? n
Done (log saved to /var/log/crmsh/ha-cluster-bootstrap.log)
```

13.2 Example cluster configuration

The following example shows a complete crm configuration for a two-node cluster (suse01, suse02) and a replicated SAP HANA database with SID HA1 and instance number 10. Priority fencing prefers the HANA primary in case of split-brain. The stand-alone database has SID QAS and instance number 20. The virtual IP address in the example is 192.168.1.20.

```
node suse01
node suse02

primitive rsc_SAPHanaTopology_HA1_HDB10 ocf:suse:SAPHanaTopology \
  op monitor interval=10 timeout=300 \
  op start interval=0 timeout=300 \
  op stop interval=0 timeout=300 \
  params SID=HA1 InstanceNumber=10

primitive rsc_SAPHana_HA1_HDB10 ocf:suse:SAPHana \
  op monitor interval=61 role=Slave timeout=700 \
  op start interval=0 timeout=3600 \
  op stop interval=0 timeout=3600 \
  op promote interval=0 timeout=3600 \
  op monitor interval=60 role=Master timeout=700 \
  params SID=HA1 InstanceNumber=10 PREFER_SITE_TAKEOVER=false \
  DUPLICATE_PRIMARY_TIMEOUT=7200 AUTOMATED_REGISTER=false \
  meta priority=100

primitive rsc_SAP_QAS_HDB20 ocf:heartbeat:SAPInstance \
  params InstanceName="QAS_HDB20_sapqasdb" \
  MONITOR_SERVICES="hdbindexserver|hdbnameserver" \
  START_PROFILE="/usr/sap/QAS/SYS/profile/QAS_HDB20_sapqasdb" \
  op start interval=0 timeout=600 \
  op monitor interval=120 timeout=700 \
  op stop interval=0 timeout=300

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
```

```

ms msl_SAPHana_HA1_HDB10 rsc_SAPHana_HA1_HDB10 \
    meta clone-max=2 clone-node-max=1 interleave=true

clone cln_SAPHanaTopology_HA1_HDB10 rsc_SAPHanaTopology_HA1_HDB10 \
    meta clone-node-max=1 interleave=true

location loc_QAS_never_on_suse01 rsc_SAP_QAS_HDB20 -inf: suse01

colocation col_QAS_never_with_HA1ip -inf: rsc_SAP_QAS_HDB20:Started \
    rsc_ip_HA1_HDB10

order ord_QASstop_before_HA1-promote Mandatory: rsc_SAP_QAS_HDB20:stop \
    msl_SAPHana_HA1_HDB10:promote

colocation col_saphana_ip_HA1_HDB10 3000: \
    rsc_ip_HA1_HDB10:Started msl_SAPHana_HA1_HDB10:Master

order ord_SAPHana_HA1_HDB10 Optional: \
    cln_SAPHanaTopology_HA1_HDB10 msl_SAPHana_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=3

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

```

13.3 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 {

    # Enable and configure quorum subsystem (default: off)
    # see also corosync.conf.5 and votequorum.5
    provider: corosync_votequorum
    expected_votes: 2
    two_node: 1
}

```

13.4 Examples for alternate STONITH methods

13.4.1 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

```

13.4.2 Example for the IPMI STONITH method

```

primitive rsc_suse01_stonith stonith:external/ipmi \
    params hostname="suse01" ipaddr="192.168.1.101" userid="stonith" \
    passwd="klllm3" 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="klllm3" 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

```

13.4.3 srCostOptMemConfig



Note

This hook script must not be used for new installations. It is listed here to document former setups.

```
""
HA/DR hook srCostOptMemConfig for method srPostTakeover()

This hook is used when deploying a "Cost Optimized Scenario".
It makes sure to reconfigure the primary database after a takeover.

The following changes to global.ini are needed to activate this hook.

[ha_dr_provider_srCostOptMemConfig]
provider = srCostOptMemConfig
path = /hana/shared/srHook/
execution_order = 2

For all hooks, 0 must be returned in case of success.

Set the following variables:
* dbinst Instance Number [for example 00 - 99]
* dbuser Username [for example SYSTEM]
* dbpwd
* user password [for example SLES4sap]
* dbport port where db listens for SQL connections [for example 30013 or 30015]
""
#
# parameter section
#
dbuser="SYSTEM"
dbpwd="<yourPassword1234>"
dbinst="00"
dbport="30013"
#
# prepared SQL statements to remove memory allocation limit
# and pre-load of column tables
#
stmt1 = "ALTER SYSTEM ALTER CONFIGURATION ('global.ini','SYSTEM') UNSET
('memorymanager','global_allocation_limit') WITH RECONFIGURE"
stmt2 = "ALTER SYSTEM ALTER CONFIGURATION ('global.ini','SYSTEM') UNSET
('system_replication','preload_column_tables') WITH RECONFIGURE"
#
```

```

# loading classes and libraries
#
import os, time
from hdbcli import dbapi
from hdb_ha_dr.client import HADRBase, Helper
#
# class definition srCostOptMemConfig
#
class srCostOptMemConfig(HADRBase):
    def __init__(self, *args, **kwargs):
        # delegate construction to base class
        super(srCostOptMemConfig, self).__init__(*args, **kwargs)

    def about(self):
        return {"provider_company" : "<customer>",
                "provider_name" : "srCostOptMemConfig", # provider name = class name
                "provider_description" : "Replication take-over script to set parameters to
default.",
                "provider_version" : "1.0"}

    def postTakeover(self, rc, **kwargs):
        """Post take-over hook."""
        self.tracer.info("%s.postTakeover method called with rc=%s" %
(self.__class__.__name__, rc))
        if rc == 0:
            # normal take-over succeeded
            conn = dbapi.connect('localhost',dbport,dbuser,dbpwd)
            cursor = conn.cursor()
            cursor.execute(stmt1)
            cursor.execute(stmt2)
            return 0
        elif rc == 1:
            # waiting for force take-over
            conn = dbapi.connect('localhost',dbport,dbuser,dbpwd)
            cursor = conn.cursor()
            cursor.execute(stmt1)
            cursor.execute(stmt2)
            return 0
        elif rc == 2:
            # error, something went wrong
            return 0

```

14 Legal notice

Copyright © 2006–2022 SUSE LLC and contributors. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or (at your option) version 1.3; with the Invariant Section being this copyright notice and license. A copy of the license version 1.2 is included in the section entitled "GNU Free Documentation License".

SUSE, the SUSE logo and YaST are registered trademarks of SUSE LLC in the United States and other countries. For SUSE trademarks, see <https://www.suse.com/company/legal/> .

Linux is a registered trademark of Linus Torvalds. All other names or trademarks mentioned in this document may be trademarks or registered trademarks of their respective owners.

Documents published as part of the SUSE Best Practices series have been contributed voluntarily by SUSE employees and third parties. They are meant to serve as examples of how particular actions can be performed. They have been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. SUSE cannot verify that actions described in these documents do what is claimed or whether actions described have unintended consequences. SUSE LLC, its affiliates, the authors, and the translators may not be held liable for possible errors or the consequences thereof.

Below we draw your attention to the license under which the articles are published.

15 GNU Free Documentation License

Copyright © 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

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. Secondly, 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.