Red Hat Enterprise Linux for SAP Solutions 8

Automating SAP HANA Scale-Out System Replication using the RHEL HA Add-On

Last Updated: 2024-10-16

Legal Notice

Abstract

This document describes how to plan and implement automated takeover for SAP HANA Scale-Out deployments.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code and documentation. We are beginning with these four terms: master, slave, blacklist, and whitelist. Due to the enormity of this endeavor, these changes will be gradually implemented over upcoming releases. For more details on making our language more inclusive, see our CTO Chris Wright's message.

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. Let us know how we can improve it.

Submitting feedback through Jira (account required)

1. Make sure you are logged in to the Jira website.
2. Provide feedback by clicking on this link.
3. Enter a descriptive title in the **Summary** field.
4. Enter your suggestion for improvement in the **Description** field. Include links to the relevant parts of the documentation.
5. If you want to be notified about future updates, please make sure you are assigned as **Reporter**.
6. Click **Create** at the bottom of the dialogue.

CHAPTER 1. INTRODUCTION

This document provides information on planning and implementing automated takeover for SAP HANA Scale-Out System Replication deployments. SAP HANA System Replication in this solution provides continuous synchronization between two SAP HANA databases to support high availability and disaster recovery. The challenges of real implementations are typically more complex than can be covered in upfront testing. Please ensure that your environment is tested extensively.

Red Hat recommends contracting a certified consultant familiar with both SAP HANA and the Pacemaker-based RHEL High Availability Add-On to implement the setup and subsequent operation.

As SAP HANA takes on a central function as the primary database platform for SAP landscapes, requirements for stability and reliability increase dramatically. Red Hat Enterprise Linux (RHEL) for SAP Solutions meets those requirements by enhancing native SAP HANA replication and failover technology to automate the takeover process. During a failover in a SAP HANA Scale-Out System Replication deployment, a system administrator must manually instruct the application to perform a takeover to the secondary environment in case there is an issue in the primary environment.

To automate this process Red Hat provides a complete solution for managing SAP HANA Scale-Out System Replication based on the RHEL HA Add-On that is part of the RHEL for SAP Solutions subscription. This documentation provides the concepts, planning, and high-level instructions on how to set up an automated SAP HANA Scale-Out System Replication solution using RHEL for SAP Solutions. This solution has been extensively tested and is proven to work, but the challenges of a real implementation are typically more complex than what this solution can cover. Red Hat therefore recommends that a certified consultant familiar with both SAP HANA and the Pacemaker-based RHEL High Availability Add-On sets up and subsequently services such a solution.

For more information about RHEL for SAP Solutions, see Overview of Red Hat Enterprise Linux for SAP Solutions Subscription.

This solution is for experienced Linux Administrators and SAP Certified Technology Associates. The solution contains planning and deployment information for SAP HANA Scale-Out with System Replication, as well as information on Pacemaker integration with RHEL 8 or later.

Building an SAP HANA scale-out environment with HANA System Replication and Pacemaker connectivity combines several complex technologies. This document contains references to SAP Notes or documentation that explains SAP HANA configuration.

An SAP HANA system as a scale-out cluster primarily extends a growing SAP HANA landscape with new hardware easily. For this feature, essential components of the infrastructure, such as storage and network, require the use of shared resources. Based on this configuration, it is possible to extend the availability of the environment by using standby nodes, providing another level of High Availability solution before a site takeover is initiated.

The SAP HANA scale-out solution can be extended to include two or more completely independent scale-out solutions that act as additional mirrors. The system replication process mirrors databases according to the active/passive method with maximum performance. The communication takes place entirely over the network. Additional infrastructure components are not needed.

Pacemaker automates the system replication process when critical components fail. For this purpose, data from the scale-out environment as well as from the system replication process are evaluated to ensure continued operation. The cluster manages the primary IP address that the client uses to connect to the database. This ensures that in the event of the cluster triggering a database takeover, the clients can still connect to the active instance.

1.1. SUPPORTING RESPONSIBILITIES

For SAP HANA appliance setups, SAP, hardware partners /cloud providers support the following:

- Supported hardware and environments
- SAP HANA
- Storage configuration
- SAP HANA Scale-Out configuration (SAP cluster setup)
- SAP HANA System Replication (SAP cluster setup)

Red Hat supports the following:

- Basic OS configuration for running SAP HANA on RHEL, based on SAP guidelines
- RHEL HA Add-On
- Red Hat HA solutions for SAP HANA Scale-Out System Replication

For more information, see SAP HANA Master Guide - Operating SAP HANA - SAP HANA Appliance - Roles and Responsibilities. For TDI setups, take a look at SAP HANA Master Guide - Operating SAP HANA - SAP HANA Tailored Data Center Intergration.

1.2. SAP HANA SCALE-OUT

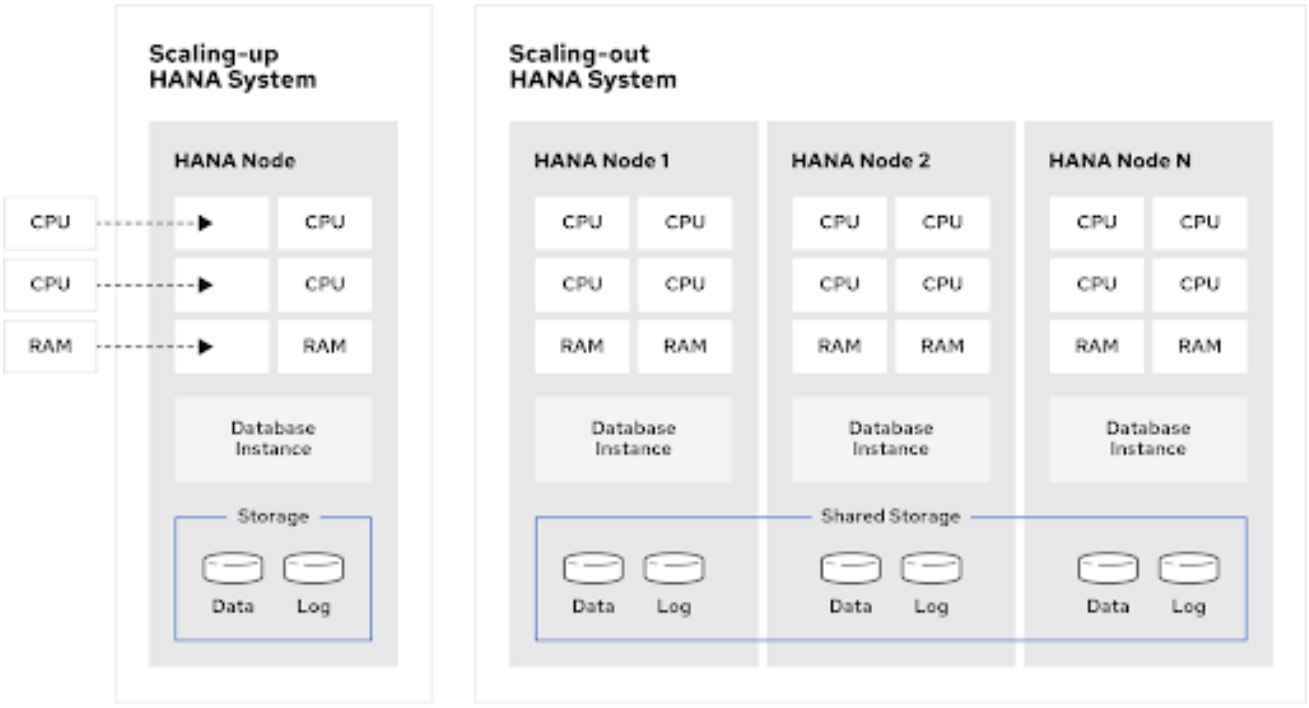
The process of scaling SAP HANA is very dynamic. During the initial setup of a server instance of a scale-up SAP HANA database, the system can be extended by additional CPUs and memory. If this expansion level is no longer sufficient, SAP extends the environment to a scale-out environment. With a properly prepared infrastructure, additional server instances can be added to the database.

NOTE

To “scale-out”, add SAP HANA database 1-n server to an existing single node database. Currently, all nodes have to be the same size in terms of CPU and RAM. The configuration of all replicated database sites has to be the same. So you have to upgrade the number of HANA nodes first on all sites before you resync the database.

The prerequisite is shared storage and a corresponding network connection for all nodes. The shared storage is used to exchange data and to use standby nodes, which can take over the functionality of existing nodes in the event of a failure.

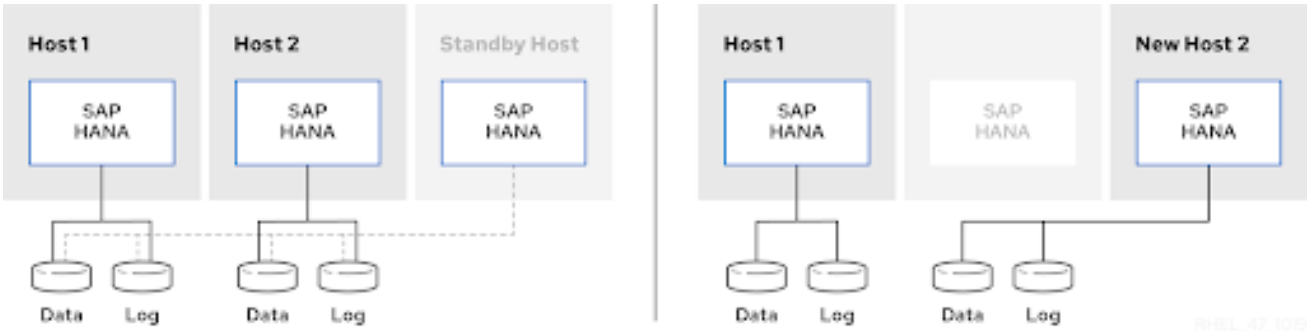
Figure 1: Overview scale-up and scale-out systems



Master nameserver

A HANA Scale-Out environment has a master configuration that defines a running master instance on one of the nodes. These master instances are the primary contact for the application server. Up to three master roles can be defined for a scale-out high-availability configuration. The master roles are switched automatically if a failure occurs. This master configuration is compatible with the standby host configuration, in which a failed host can take over the tasks of a failed master node.

Figure 2: Scale-out functionality of the used storage



1.3. SCALE-OUT STORAGE CONFIGURATION

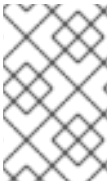
Scale-out storage configuration allows SAP HANA to be flexible in the scale-out environment and to dynamically move the functionality of the nodes in the event of a failure. Since the data is made available to all nodes, the SAP instances only have to be ready to take over the process of the failed components.

There are two different shared storage scenarios for SAP HANA scale-out environments:

- The first scenario is shared file systems, which offer a file system of all directories over NFS or IBM's GPFS. In this scenario, the data is available on all nodes, all the time.
- The second scenario is non-shared storage, which is used to exclusively integrate the required data when needed. All data is managed over the SAP HANA storage connector API, and it

removes access from nodes using the appropriate mechanisms, for example, SCSI 3 reservations.

For both scenarios, ensure that the /**hana**/**shared** directory is made available as a shared file system. This directory must be available and shared independently of the scenarios.



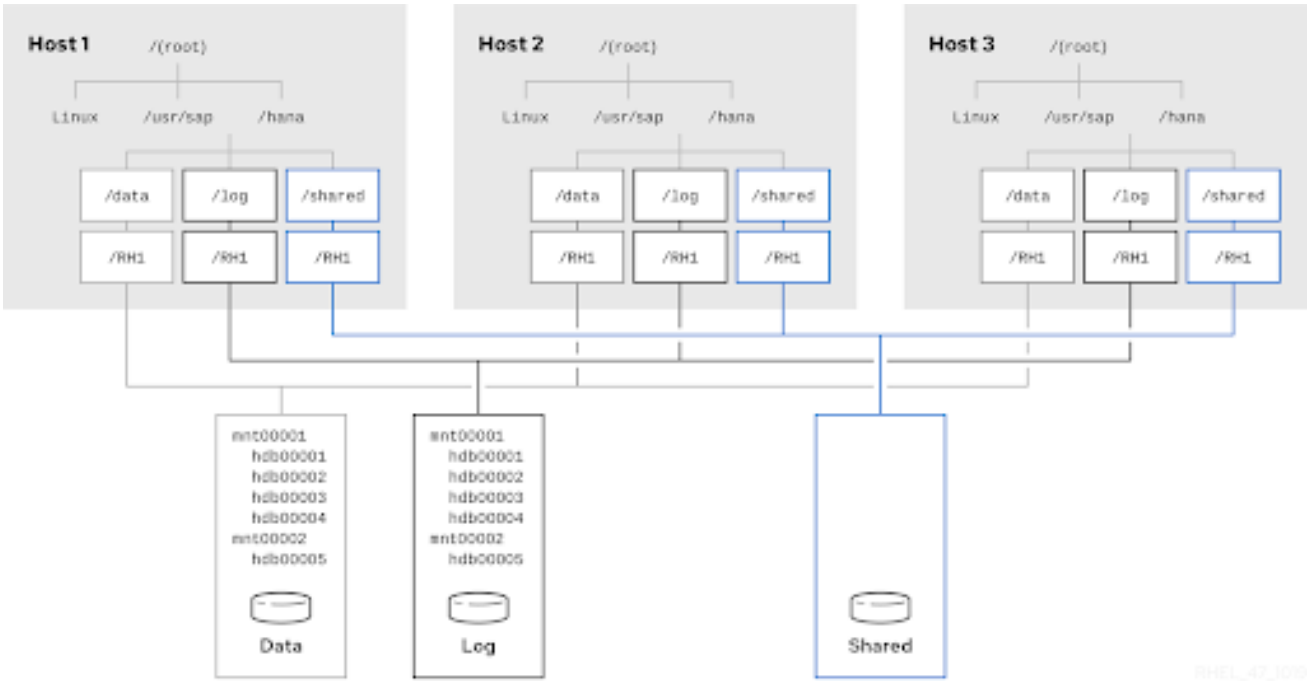
NOTE

If you want to monitor these shared file systems, you can optionally create file system resources. The entries in the /**etc**/**fstab** should be removed; the mount is only managed by the file system resources.

1.3.1. Shared storage

Shared file systems deliver the required data on every host. When configured, SAP HANA accesses the necessary data. The data can be shared easily because the shared directories are mounted on all nodes. The installation proceeds as normal after deployment. SAP HANA has access to all directories: /**hana**/**data**, /**hana**/**log** and /**hana**/**shared**.

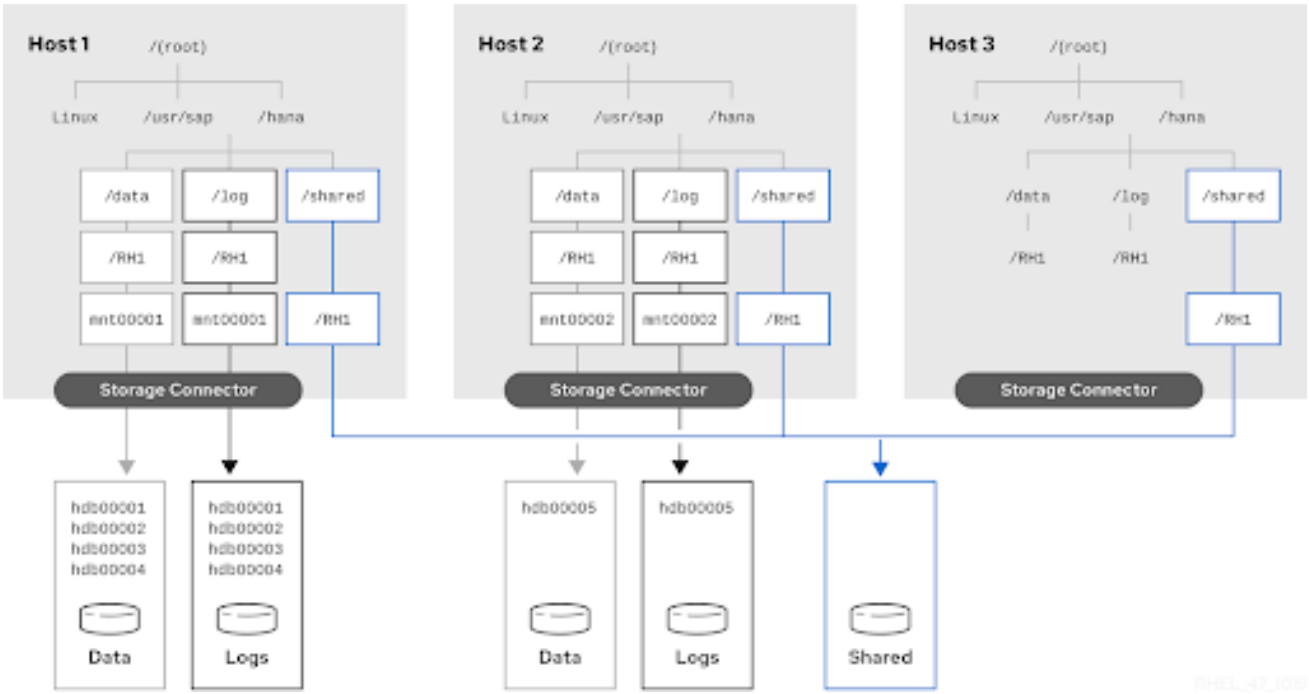
Figure 3: Functionality and working paths of the scale-out process with shared storage



1.3.2. Non-shared storage

A non-shared storage configuration is more complex than a shared storage configuration. It requires a supported storage component and an individual configuration of the storage connector in the SAP HANA installation process. The SAP HANA database reconfigures the RHEL systems with several internal changes, for example, sudo access, lvm, or multipath. With every change of the node definition, SAP HANA is changing access to the storage directly over SCSI3 reservations. The non-shared storage configuration is more optimised than the shared storage configuration because it has direct access to the storage system.

Figure 4: Functionality and working paths of the scale-out process with the storage connector



1.4. SAP HANA SYSTEM REPLICATION

SAP HANA System Replication provides a way for its SAP HANA environment to replicate the database across multiple sites. The network replicates the data and preloads it into the second SAP HANA installation. SAP HANA System Replication significantly reduces recovery time in case there is a failure of the primary HANA Scale-Out site. You must ensure that all replicated environments are built with identical specifications across hardware, software, and configuration settings.

1.5. NETWORK CONFIGURATION

Three networks are the minimum network requirements for an SAP HANA Scale-Out System Replication setup that is managed by the RHEL HA Add-On. Nevertheless, an SAP-recommended network configuration should be used to build up a high performing production environment.

The three networks are:

- Public network: Required for the connection of the application server and clients (minimum requirement).
- Communication network: Required for system replication communication, internode communication, and storage configuration.
- Heartbeat network: Required for HA cluster communication.

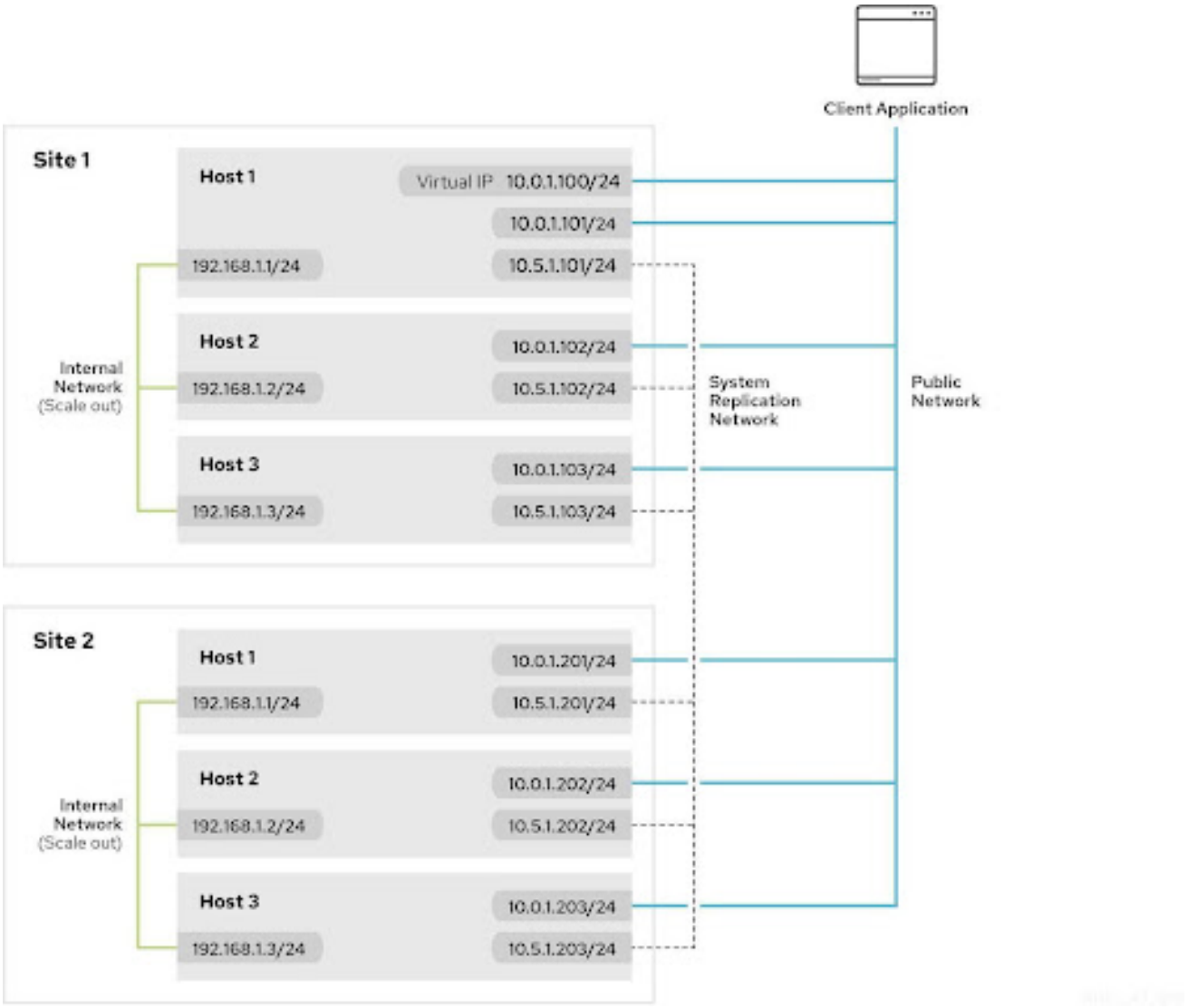
The recommended configuration is designed with the following networks:

- Application server network
- Client network
- Replication network
- Storage network
- Two internode networks

- Backup network
- Admin network
- Pacemaker network

Based on the configuration of this solution, changes in the SAP HANA configuration process are required. The system replication hostname resolution is adjusted to the network that is used for the system replication. This is described in the SAP HANA Network Requirements documentation.

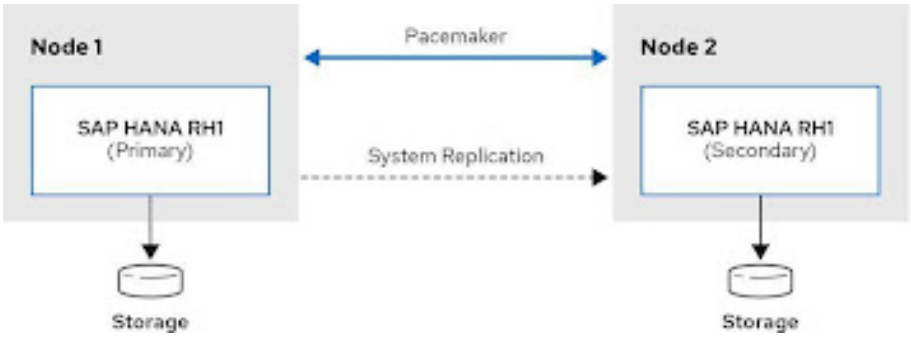
Figure 5: Example Network configuration of two scale-out systems connected over SAP HANA system replication



1.6. RHEL HA ADD-ON

In the solution described in this document, the RHEL HA Add-On is used for ensuring the operation of SAP HANA Scale-Out System Replication across two sites. For this reason, resource agents published specifically for SAP HANA scale-out environments are used, which manage the SAP HANA Scale-Out System Replication environment. Based on the current status of the SAP HANA Scale-Out System Replication environment, a decision can be made to either switch the active master node to another available standby node or to switch the entire active side of the scale-out system replication environment to the second site. For this solution, a fencing mechanism is configured to avoid split-brain constellations.

Figure 6: Overview of Pacemaker integration based on a system replication environment



For more information about using the RHEL HA Add-On to set up HA clusters on RHEL 8, see the following documentation:

- Configuring and managing high availability clusters
- Support Policies for RHEL High Availability Clusters

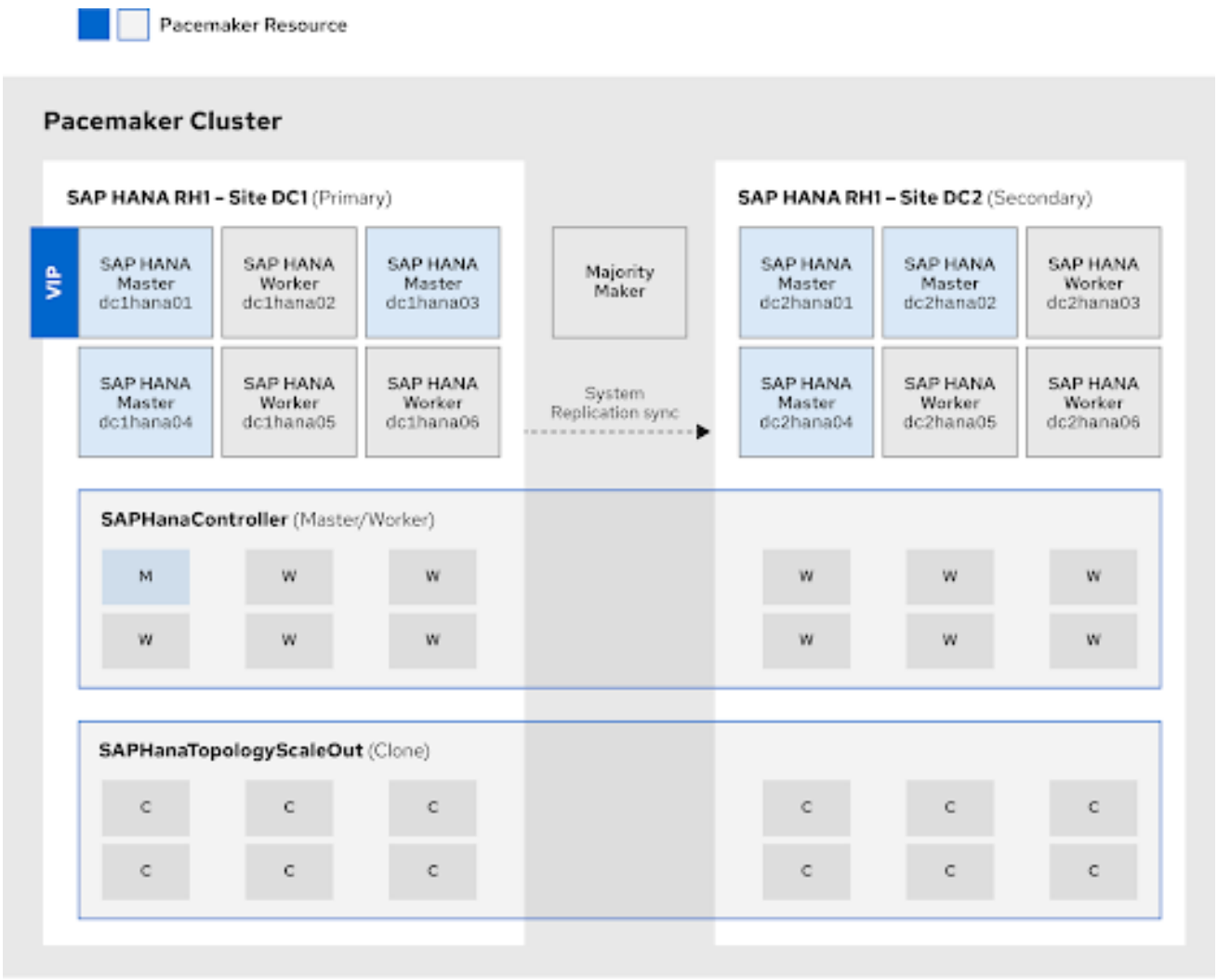
It is important to understand scale-out and system replication methods from the SAP HANA database because SAP HANA scale-out resource agents are using data from every environment.

At first, the resource agent is watching for a stable scale-out environment on every site. It checks if enough SAP HANA scale-out master nameserver nodes are configured and in a valid state. Subsequently, the resource agent checks the system replication state. If everything is working correctly, it attaches the virtual IP address to the active master node on the master site of the system replication. In a failure state, the cluster is configured to switch the system replication configuration automatically.

The definition of a failure state is dependent on the configuration of the master nameserver. For example, when one master nameserver is configured, the cluster switches directly to the other datacenter if the master node fails. If up to three master nameservers are configured, the SAP HANA environment heals itself before switching to the other datacenter. Pacemaker is working with the scoring numbers to make decisions on what should be done. When running SAP HANA, it is very important that these parameters are not changed in a cluster setup.

Pacemaker configuration is also based on fencing configuration that uses Shoot The Other Node In The Head (STONITH). An unresponsive node does not mean that it is not accessing data. Use STONITH to fence the node and be sure that data is safe. STONITH protects data from being corrupted by rogue nodes or concurrent access. If the communication between the two sites is lost, both sites may believe they are able to continue working, which can cause data corruption. This is also called a split-brain scenario. To prevent this, a quorum can be added, which helps to decide who is able to continue. A quorum can either be an additional node or a qdevice. In our example, we are using the additional node **majoritymaker**.

Figure 7: Example of system replication with scale out



1.7. RESOURCE AGENTS

The cluster configuration is working with two resource agents.

1.7.1. SAPHanaTopology resource agent

The **SAPHanaTopology** resource agent is a cloned resource that receives all of its data from the SAP HANA environment. A configuration process in SAP HANA called “system replication hook" generates this data. Based on this data, the resource agent calculates the Pacemaker scoring for the Pacemaker service. The scoring is used by the cluster to decide if it should initiate switching the system replication from one site to the other. If the scoring value is higher than a predefined value, the cluster switches the system replication.

1.7.2. SAPHanaController resource agent

The **SAPHanaController** resource agent controls the SAP HANA environment and executes all commands for an automatic switch, or it changes the active site of the system replication.

CHAPTER 2. PREPARING THE SAP HANA SCALE-OUT ENVIRONMENT

For a complete SAP HANA Scale-Out environment with System Replication and Pacemaker integration, it is advisable to gather all necessary data in advance and to prepare the infrastructure for the installation process. The installation of SAP HANA requires a large number of variables from different operating system components, including SAP itself. The minimum requirements are described in this chapter.

2.1. SUBSCRIPTIONS AND REPOSITORIES

Requirements for SAP HANA deployment:

- RHEL for SAP Solutions Subscriptions must be enabled on all RHEL servers running SAP HANA.
- Staging environment with satellite server to ensure the correct package versions are installed on every system.The following repository must be enabled for installing SAP HANA on RHEL 8:
- **rhel-8-SAP-Solutions**:
  - rhel-8-for-<arch>-sap-solutions-rpms (RHEL 8.10)
  - rhel-8-for-<arch>-sap-solutions-e4s-rpms (RHEL 8.0 to 8.8)

The **<arch>** denotes the specific hardware architecture as follows:

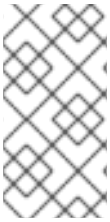
- x86_64
- ppc64le

For more information, see Overview of Red Hat Enterprise Linux for SAP Solutions Subscription and RHEL for SAP Subscriptions and Repositories.

A separate storage network, backup network, and admin network are not required for this solution. In addition to the network configuration, use Pacemaker to configure an additional virtual IP. This IP address allows SAP application servers and certain end-users to communicate with the SAP HANA environment.

The following example lists the minimum requirements for a network configuration of eight SAP HANA nodes.

Parameter	Value
domainname	example.com
NTP Server 1	0.de.pool.ntp.org
NTP Server 2	1.de.pool.ntp.org
Virtual IP	10.111.222.52/24



NOTE

Note: Pacemaker manages the virtual IP (VIP) address in the public network for communication between the SAP application server and the SAP HANA database. The following example lists the physical addresses that are mapped to hosts with three NICs (Network Interface Cards).

Table 2.1. Datacenter 1

Hostname	Public Network	HANA Communication	Pacemaker
dc1hana01	10.0.1.21/24	192.168.101.101/24	192.168.102.101/24
dc1hana02	10.0.1.22/24	192.168.101.102/24	192.168.102.102/24
dc1hana03	10.0.1.23/24	192.168.101.103/24	192.168.102.103/24
dc1hana04	10.0.1.24/24	192.168.101.104/24	192.168.102.104/24

Table 2.2. Datacenter 2

Hostname	Public Network	HANA Communication	Pacemaker
dc2hana01	10.0.1.31/24	192.168.101.201/24	192.168.102.201/24
dc2hana02	10.0.1.32/24	192.168.101.202/24	192.168.102.202/24
dc2hana03	10.0.1.33/24	192.168.101.203/24	192.168.102.203/24
dc2hana04	10.0.1.34/24	192.168.101.204/24	192.168.102.204/24

Hostname	Public Network	Pacemaker
majoritymaker	10.0.1.41/24	192.168.102.100/24

2.2. STORAGE

There are two methods to configure storage for an SAP HANA Scale-Out scenario:

- Shared storage
- Non-shared storage

There is no communication between both scale-out environments on the storage level. As a result, storage configuration must be completed on each scale-out environment to ensure SAP HANA System Replication is working as expected.

2.2.1. Shared storage

Shared storage configuration requires methods and mount points. When configuring shared storage over NFS, the NFS Server, NFS Paths, and directories are necessary, and it is necessary to provide the directories **/hana/data**, **/hana/log**, and **/hana/shared**.

Table 2.3. Datacenter 1

Method	NFS Server	NFS Path	Mount Point
NFS	10.0.1.61/24	/data/dc1/data	/hana/data
NFS	10.0.1.61/24	/data/dc1/log	/hana/log
NFS	10.0.1.61/24	/data/dc1/shared	/hana/shared

Table 2.4. Datacenter 2

Method	NFS Server	NFS Path	Mount Point
NFS	10.0.1.61/24	/data/dc2/data	/hana/data
NFS	10.0.1.61/24	/data/dc2/log	/hana/log
NFS	10.0.1.61/24	/data/dc2/shared	/hana/shared

2.2.2. Non-shared storage

Non-shared storage configuration requires the integration of the storage connector. The storage connector manages access to the LUNs or LVM Devices over SCSI or LVM locking mechanisms. For this configuration type, WWID or LVM devices are needed. For a non-shared storage configuration, one shared directory is required for each scale-out environment. This configuration is described in the SAP HANA Fiber Channel Storage Connector Admin Guide.

Table 2.5. Non-shared storage example

Parameter	Value
ha_provider	hdb_ha.fcClient

Table 2.6. Datacenter 1

Method	Parameter Name	WWID
SAN	partition_1_data wwid	3600508b400105e210000900000491000
SAN	partition_1_log wwid	3600508b400105e210000900000492000

SAN	partition_2_data wwid	3600508b400105e210000900000493000
SAN	partition_2_log wwid	3600508b400105e210000900000494000
SAN	partition_3_data wwid	3600508b400105e210000900000495000
SAN	partition_3_log wwid	3600508b400105e210000900000496000

Table 2.7. Datacenter 2

Method	Parameter Name	WWID
SAN	partition_1_data wwid	3600508b400105e210000900000491000
SAN	partition_1_log wwid	3600508b400105e210000900000492000
SAN	partition_2_data wwid	3600508b400105e210000900000493000
SAN	partition_2_log wwid	3600508b400105e210000900000494000
SAN	partition_3_data wwid	3600508b400105e210000900000495000
SAN	partition_3_log wwid	3600508b400105e210000900000496000

2.2.3. Shared devices

Table 2.8. Datacenter 1

Method	NFS Server	NFS Path	Mount Point
NFS	10.0.1.61	/data/dc1/shared	/hana/shared

Table 2.9. Datacenter 2

Method	NFS Server	NFS Path	Mount Point

NFS	10.0.1.61	/data/dc2/shared	/hana/shared

NOTE

If the shared storage is managed by a filesystem resource, the mount should not be added into the **/etc/fstab**.

2.3. SAP HANA

There are four steps in building an SAP HANA deployment for a scale-out environment with SAP HANA System Replication:

1. Configuring the operating system.
2. Installing the SAP Host Agent.
3. Deploying two scale-out environments.
4. Activating HANA System Replication after both SAP HANA Scale-Out environments are running.

Preparation of the RHEL environment includes provisioning the SAP HANA installation sources. Installation sources are available from SAP. You must have an SAP account to download the installation sources, which are provided over a shared directory, or copied manually on every host.

SAP software can be downloaded from the SAP Software Center. In our example, we put the software in a shared directory /install.

Table 2.10. Installation sources for SAP HANA Host Agent

Software	Path
Host Agent	/install/51053381/DATA_UNITS/HDB_SERVER_LINUX_X86_64/server/HOSTAGEN T.TGZ
SAP HANA	/install/51053381/DATA_UNITS/HDB_SERVER_LINUX_X86_64/

The following information is required for the deployment of the SAP HANA Host Agent:

Parameter	Value
sapadm user password	Us3Your0wnS3cur3Password
Hostagent SSL certificate password	Us3Your0wnS3cur3Password
sapadm User ID	996

For an SAP HANA deployment, the following additional parameters are required:

Parameter	Value
shmadm group ID	20201
sapsys group ID	996
SID	RH1
System number	10
<sid>adm password	Us3Your0wnS3cur3Password
HANA components	client,server
System type	Master
System usage	custom
System User Password (SAP)	Us3Your0wnS3cur3Password

Table 2.11. SAP HANA deployment datacenter 1

hdblcm Parameter	Value
hostname	dc1hana01
Addhosts parameter	dc1hana02:role=worker,dc1hana03:role=worker,dc1hana04:role=standby
ScaleOut Network DC1 internal_network	192.168.101.0/24

Table 2.12. SAP HANA deployment datacenter 2

hdblcm Parameter	Value
hostname	dc2hana01
Addhosts parameter	dc2hana02:role=worker,dc2hana03:role=worker,dc2hana04:role=standby
ScaleOut Network DC2 internal_network	192.168.101.0/24

Table 2.13. HSR configuration

Parameter	Value

Operation mode	logrelay
Replication mode	sync
Backup directory	/hana/shared/L01/HDB10/backup/

Table 2.14. Datacenter 1

Parameter	Value
System replication name	DC1
HSR type	PRIMARY
HSR remote host	dc1hana01
SR network	192.168.101.0/24

Table 2.15. Datacenter 2

Parameter	Value
System replication name	DC2
HSR type	Secondary
HSR remote host	dc2hana01
SR network	192.168.101.0/24

2.4. PACEMAKER

Pacemaker manages the configuration of SAP HANA Scale-Out System Replication. For a working configuration, Pacemaker requires a fencing method. This can be achieved by the STONITH Pacemaker configuration. For an overview of STONITH methods, refer to Support Policies for RHEL High Availability Clusters-Fencing/STONITH. There are many fence-agents available, please also check:

```
yum search fence-agents
```

Pacemaker fencing configuration is dependent on the underlying hardware or the virtualization environment. In this solution, because Red Hat Virtualization (RHV) is used, the **fence_rhevm** fencing method must be configured according to the environment.

To prevent split-brain scenarios a quorum is required. which is realized using an additional cluster node **majoritymaker**. If you need more information about quorum, please check Design Guidance for RHEL High Availability Clusters - Considerations with qdevice Quorum Arbitration and Exploring Concepts of RHEL High Availability Clusters-Quorum.

Table 2.16. Majoritymaker parameters

Hostname	Public Network	Pacemaker
Majoritymaker	10.0.1.42/24	192.168.101.100/24

Table 2.17. General parameters

Parameter	Value
Cluster name	hana-scaleout-sr
Fencing method	fence_rhevm
Fencing ipaddr/hostname	10.20.30.40
Fencing parameter	fencing_user/password
Corosync network	192.168.101.0/24
Password hacluster user	Us3Your0wnS3cur3Password

In this example, we use **fence_rhevm**. For more details about configuring **fence_rhevm**, please check How do I configure a fence_rhevm stonith device in a Red Hat High Availability cluster? .

CHAPTER 3. CONFIGURING THE SAP HANA SCALE-OUT ENVIRONMENT

This solution is about setting up and configuring an SAP HANA Scale-Out environment with System Replication and Pacemaker. It is separated into two parts: Setting up a basic RHEL configuration, which is different for every environment. Deploying and configuring SAP HANA Scale-Out for System Replication and Pacemaker.

The minimal requirement is using 2 nodes per site plus a quorum device which is in our example an additional **majoritymaker** node. The test environment described here is built up with eight SAP HANA nodes and an additional **majoritymaker** node for cluster quorum. All SAP HANA nodes have a 50 GB root disk and an additional 80 GB partition for the **/usr/sap** directory. Every SAP HANA node has 32 GB RAM. The **majoritymaker** node can be smaller, for example 50GB root disk and 8GB of RAM. For the shared directories, there are two NFS pools with 128 GB. To ensure a smooth deployment, it is recommended that you record all required parameters as described in the Preparing the SAP HANA Scale-Out environment section of this document. The following example provides an overview of the required configuration parameters.

Environment

<table>
<tr><td colspan="3">Pacemaker</td></tr>
<tr><td>4 Nodes (3 + 1)</td><td>Majoritymaker</td><td>4 Nodes (3 + 1)</td></tr>
<tr><td>Shared Storage (NFS for DC1)</td><td>← System Replication →</td><td>Shared Storage (NFS for DC2)</td></tr>
<tr><td>Network
Public Network
HANA Network
Cluster Network</td><td></td><td>Network
Public Network
HANA Network
Cluster Network</td></tr>
</table>

3.1. SETTING UP A BASIC RHEL CONFIGURATION

Use the procedures in this section to set up a basic RHEL configuration in your environment. You can also check for RHEL 8 in SAP-Notes 2772999 - Red Hat Enterprise Linux 8.x: Installation and Configuration and 2777782 - SAP HANA DB: Recommended OS Settings for RHEL 8 .

Please check SAP Note 2235581 - SAP HANA: Supported Operating Systems to verify that the RHEL 8 minor release that is going to be used is supported for running SAP HANA. In addition, it is also necessary to check with the server/storage vendor or cloud provider to make sure that the combination of SAP HANA and RHEL 8 is supported on the servers/storage or cloud instances that are to be used.

For information about the latest RHEL release, see the Release Notes document available on the Customer Portal. To find your installed version and see if you need to update, run the following command:

```
[root:~]# subscription-manager release
Release: 8.2
[root:~]# cat /etc/redhat-release Red Hat Enterprise Linux
```

```
release 8.2 (Ootpa)
[root:~]#
```

3.1.1. Registering your RHEL system and enabling repositories

NOTE

- In this solution, Red Hat receives system registration directly as there is no staging configuration. You are recommended to create a staging configuration for SAP HANA systems to have a reproducible environment. Satellite Server provides packet management, which also includes the staging process (**dev**/**qa**/**prod**.) For more information, refer to the Satellite Server product information.
- You must verify that the hostname is correct before registering the system, as this makes it easier to identify systems when managing subscriptions. For more information, refer to the solution How to set the hostname in Red Hat Enterprise Linux 7, 8, 9. For RHEL 8, check Configuring basic system settings.

Prerequisites

- RHEL 8 is installed.
- You are logged in as user root on every host, including the 'majoritymaker` for Subscription Management.

Procedure

1. If a staging configuration is not present, you can assign the registration of the SAP HANA test deployment directly to Red Hat Subscription Management (RHSM) with the following command:

   ```
   [root:~]# subscription-manager register
   ```

2. Enter the username and password.
3. List all pools available with the **rhel-8-for-x86_64-sap-solutions-rpms** repositories:

   ```
   [root:~]# subscription-manager list --available --matches="rhel-8-for-x86_64-sap-solutions-rpms"
   ```

   For more information, refer to Configuring basic system settings.

   NOTE

   The company pool ID is required. If the list is empty contact Red Hat for a list of the company's subscriptions.

4. Attach the pool ID to your server instances:

   ```
   [root:~]# subscription-manager attach --pool=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   ```

5. Check if the repo for **sap-solutions** is enabled:

```
[root:~]# yum repolist | grep sap-solution
rhel-8-for-x86_64-sap-solutions-rpms RHEL for x86_64 - SAP Solutions (RPMs)
```

You can enable the RHEL 8 required repos:

```
[root:~]# subscription-manager repos --enable=rhel-8-for-x86_64-sap-solutions-rpms --
enable=rhel-8-for-x86_64-highavailability-rpms
```

For more information, see RHEL for SAP Subscriptions and Repositories .

6. Update the packages on all systems to verify that the correct RPM packages and versions are installed:

```
[root:~]# yum update -y
```

3.1.2. Configuring network settings

This section describes the network parameters used in this solution. The configuration of this solution was dependent on the environment, and it should be considered an example. The configuration of the network should be done according to SAP specifications. An example for node **dc1hana01** is included in the Preparing the SAP HANA Scale-Out environment section of this document.

```
[root:~]# nmcli con add con-name eth1 ifname eth1 autoconnect yes type ethernet ip4
192.168.101.101/24 nmcli con add con-name eth2 ifname eth2 autoconnect yes type ethernet ip4
192.168.102.101/24
```

3.1.3. Configuring /etc/hosts

Use this procedure to configure /**etc**/**hosts** on your RHEL systems. This configuration is necessary for consistent hostname resolution.

Procedure

1. Login as user root on every host and configure the /**etc**/**hosts** file.
2. Create a host entry for every SAP HANA host in the scale-out environment.
3. Copy the hosts file to every node. It is important to set the hostname in the order shown in the following output example. If not, the SAP HANA environment fails during the deployment or operating process.

NOTE

This configuration is based on the parameters listed in the Preparing the SAP HANA Scale-Out environment section of this document.

```
[root:~]# cat << EOF >> /etc/hosts
10.0.1.21 dc1hana01.example.com dc1hana01
10.0.1.22 dc1hana02.example.com dc1hana02
10.0.1.23 dc1hana03.example.com dc1hana03
10.0.1.24 dc1hana04.example.com dc1hana04
10.0.1.31 dc2hana01.example.com dc2hana01
```

```
10.0.1.32 dc2hana02.example.com dc2hana02
10.0.1.33 dc2hana03.example.com dc2hana03
10.0.1.34 dc2hana04.example.com dc2hana04
10.0.1.41 majoritymaker.example.com majoritymaker
EOF
```

3.1.4. Configuring disks

Complete this procedure to configure the disks on your RHEL systems.

Procedure

1. Login as user root on every SAP HANA host for the additional **/usr/sap** partition.

NOTE

In general, the default XFS format and mount options are optimal for most workloads. Red Hat recommends that the default values are used unless specific configuration changes are expected to benefit the workload of the file system. All supported file systems can be used. For more information, refer to SAP Note 2972496 - SAP HANA Filesystem Types . If software RAID is used, the **mks.xfs** command automatically configures itself with the correct stripe unit and width to align with the hardware.

2. Create the required mount points:

```
[root:~]# mkdir -p /usr/sap
```

3. On the logical volume, create file systems based on XFS:

```
[root:~]# mkfs -t xfs -b size=4096 /dev/sdb
```

For more information about the creation of an XFS filesystem and the tuning possibilities, run the **man mkfs.xfs** command. For optimal performance of the XFS file system, refer to the article What are some of best practices for tuning XFS filesystems .

4. Write the mount directives to **/etc/fstab**:

```
[root:~]# echo "/dev/sdb /usr/sap xfs defaults 1 6" >> /etc/fstab
```

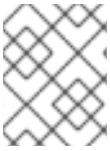
NOTE

If mount points are managed by filesystem resources, these file systems must then be later commented out again in the **/etc/fstab** file.

5. Check if XFS filesystems from **/etc/fstab** can be mounted:

```
[root:~]# mount /usr/sap
```

3.1.5. Configuring Scale-Out with shared storage for each datacenter



NOTE

In cloud environments, there can be different sources for the same mount point in different availability zones.

Use this procedure to configure scale-out with shared services for each datacenter.

Procedure

1. Login as user root on every SAP HANA host for the shared storage configuration.

NOTE

The **nfs-utils** package is required. Every datacenter requires its own storage configuration. For this example, the storage configuration is built as a shared storage environment. Both scale-out environments are using its own NFS share. This configuration is based on the information in the Preparing the SAP HANA Scale-Out environment section of this document. In a production environment, this procedure should be configured as supported by your preferred hardware vendor.

2. Install the **nfs-utils** package:

```
[root:~]# yum install -y nfs-utils
```

3. Configure the nodes in Datacenter 1:

```
[root:~]# mkdir -p /hana/{shared,data,log}
cat <<EOF >> /etc/fstab
10.0.1.61:/data/dc1/shared /hana/shared nfs4 defaults 0 0
10.0.1.61:/data/dc1/data /hana/data nfs4 defaults 0 0
10.0.1.61:/data/dc1/log /hana/log nfs4 defaults 0 0
EOF
```

To mount the volumes run the following command:

```
[root:~]# mount -a
```

4. Configure the nodes in Datacenter 2:

```
[root:~]# mkdir -p /hana/{shared,data,log}
cat <<EOF >> /etc/fstab
10.0.1.62:/data/dc2/shared /hana/shared nfs4 defaults 0 0
10.0.1.62:/data/dc2/data /hana/data nfs4 defaults 0 0
10.0.1.62:/data/dc2/log /hana/log nfs4 defaults 0 0
EOF
```

To mount the volumes run the following command:

```
[root:~]# mount -a
```

3.2. CONFIGURING AND DEPLOYING SAP HANA

3.2.1. Configuring RHEL settings required for running SAP HANA

Use this procedure to configure the HA cluster nodes for running SAP HANA. It is necessary to perform these steps on each RHEL system on which a HANA instance is running.

Prerequisites

1. You are logged in as user root on every host of the shared storage configuration.
2. You have prepared the installation source of SAP HANA.
3. You have set the hostname compatible with SAP HANA:

```
[root:~]# hostnamectl set-hostname dc1hana01
```

Procedure: Verifying /etc/hosts

1. Verify that **/etc/hosts** contains an entry matching the hostname and IP address of the system: **.example.com**:

```
[root:~]# hostname
<hostname>
[root:~]# hostname -s
<hostname>
[root:~]# hostname -f
<hostname>.example.com
[root:~]# hostname -d
example.com
```

2. Set the system language to English:

```
[root:~]# localectl set-locale LANG=en_US.UTF-8
```

Procedure: Configuring NTP

1. Edit **/etc/chrony.conf** and verify that the server lines reflect your ntp servers:

```
[root:~]# yum -y install chrony
[root:~]# systemctl stop chronyd.service
```

2. Check time server entries:

```
[root:~]# grep ^server /etc/chrony.conf
server 0.de.pool.ntp.org
server 1.de.pool.ntp.org
```

3. Enable and start the chrony service:

```
[root:~]# systemctl enable chronyd.service
[root:~]# systemctl start chronyd.service
[root:~]# systemctl restart systemd-timedated.service
```

4. Verify that the **chrony** service is enabled:

```
[root:~]# systemctl status chronyd.service
chronyd.service enabled
[root:~]# chronyc sources
210 Number of sources = 3
MS Name/IP address Stratum Poll Reach LastRx Last sample
===============================================================================
^* 0.de.pool.ntp.org 2 8 377 200 -2659ns[-3000ns] +/- 28ms
^-de.pool.ntp.org 2 8 377 135 -533us[ -533us] +/- 116ms
^-ntp2.example.com 2 9 377 445 +14ms[ +14ms] +/- 217ms
```

3.2.2. Preconfiguring RHEL for SAP HANA

Use this procedure to preconfigure the RHEL system for SAP HANA. This configuration is based on published SAP Notes. Run this procedure on every SAP HANA host in the cluster as user root.

NOTE

- This procedure is based on SAP Notes SAP Note 2777782: SAP HANA DB: Recommended OS Settings for RHEL 8 and SAP Note 2772999 - Red Hat Enterprise Linux 8.x: Installation and Configuration.
- On RHEL 8, you can also use the RHEL System Roles for SAP to automate the installation and configuration of the HA cluster nodes. More information can be found here: Red Hat Enterprise Linux System Roles for SAP .

3.2.3. Installing the SAP Host Agent

SAP Host Agent is installed automatically during the installation of all new SAP system instances or instances with SAP kernel 7.20 or higher. This manual installation is not necessary in most cases. Please install SAP HANA first and then check if the installation of **saphostagent** is still needed.

Prerequisites

- You have verified that **umask** configuration is configured as a standard value (the command umask should reply 0022.); otherwise, the SAP Host Agent installation could fail.
- You are logged in as user root on every host for SAP Host Agent installation.

NOTE

The user and group are created during the SAP HANA installation if the user/group does not exist and the **SAPHOSTAGENT** is installed/upgraded through the installation of SAP software.

Procedure (optional)

1. Create the **sapadm** and **sapsys** user for the SAP Host Agent and set the password for the **sapadm** user. The UID 996 of the user **sapadm** and the GID 79 of the group **sapsys** are based on the parameters in the Preparing the SAP HANA Scale-Out environment section of this document.

```
[root:~]# adduser sapadm --uid 996
[root:~]# groupadd sapsys --gid 79
[root:~]# passwd sapadm
```

2. Create a temp directory, unpack the installation source, and install the SAP Host Agent from the temp directory. The variable **INSTALLDIRHOSTAGENT** is an example:

```
[root:~]# export TEMPDIR=$(mktemp -d)
[root:~]# export
INSTALLDIRHOSTAGENT=/install/HANA/DATA_UNITS/HDB_SERVER_LINUX_X86_64/
[root:~]# systemctl disable abrtd
[root:~]# systemctl disable abrt-ccpp
[root:~]# cp -rp ${INSTALLDIRHOSTAGENT}/server/HOSTAGENT.TGZ $TEMPDIR/ cd
$TEMPDIR
[root:~]# tar -xzvf HOSTAGENT.TGZ
[root:~]# cd global/hdb/saphostagent_setup/
[root:~]# ./saphostexec -install
```

Secure operation only works with an encrypted connection. You can configure a working SSL connection to achieve this. An SSL password is required. The following example is based on the parameters in the Preparing the SAP HANA Scale-Out environment section of this document.

```
[root:~]# export MYHOSTNAME=$(hostname)
[root:~]# export SSLPASSWORD=Us3Your0wnS3cur3Password
[root:~]# export LD_LIBRARY_PATH=/usr/sap/hostctrl/exe/
[root:~]# export SECUDIR=/usr/sap/hostctrl/exe/sec
[root:~]# cd /usr/sap/hostctrl/exe
[root:~]# mkdir /usr/sap/hostctrl/exe/sec
[root:~]# /usr/sap/hostctrl/exe/sapgenpse gen_pse -p SAPSSLS.pse -x $SSLPASSWORD -r
/tmp/${MYHOSTNAME}-csr.p10 "CN=$MYHOSTNAME"
[root:~]# /usr/sap/hostctrl/exe/sapgenpse seclogin -p SAPSSLS.pse -x $SSLPASSWORD -O
sapadm chown sapadm /usr/sap/hostctrl/exe/sec/SAPSSLS.pse
[root:~]# /usr/sap/hostctrl/exe/saphostexec -restart*
```

3. Verify the SAP Host Agent is available for all SAP HANA nodes:

```
[root:~]# netstat -tulpen | grep sapstartsrv
tcp     0    0 0.0.0.0:50014   0.0.0.0:*  LISTEN   1002  84028   4319/sapstartsrv
tcp     0    0 0.0.0.0:50013   0.0.0.0:*  LISTEN   1002  47542   4319/sapstartsrv
```

NOTE

Not all processes are identified. Non-owned process information are not be shown. You have to be root to see all processes.

```
[root:~]# netstat -tulpen | grep 1129
tcp 0  0     0.0.0.0:1129  0.0.0.0:*  LISTEN  996  25632  1345/sapstartsrv
```

For more information about how to install SAP Host Agent, see SAP Host Agent Installation.

3.2.4. Deploying SAP HANA with Scale-Out and System Replication

Before deploying SAP HANA with Scale-Out and System Replication, you must understand SAP network mappings. This solution provides minimal configuration details for deployment in a lab environment. However, when configuring a production environment, it is necessary to map the scale-out network

communication and system replication communication over separate networks. This configuration is described in the Network Configuration for SAP HANA System Replication .

The SAP HANA database should be installed as described according to the SAP HANA Server Installation and Update Guide.

There are different options to set up the SAP HANA database. You must install the database on both datacenters with the same SID. A scale-out configuration needs at least 2 HANA instances per site.

The installation for each HANA site consists of the following steps:

- Install SAP HANA database on the first node using **hdblcm** (check **hdblcm** in the **SAP_HANA_DATABASE** subdirectory of the SAP HANA installation media).
- Configure the internal network for the scale-out configuration on this first node (this is only necessary once):

```
[root:~]# ./hdblcm --action=configure_internal_network
```

- Install the additional HANA instances on the other nodes using the shared executable created by the first installation:

```
[root:~]# /hana/shared/RH1/hdblcm/hdblcm
```

- Choose the right HANA role (worker or standby) for each HANA instance.
- Repeat the same steps for the secondary HANA site.

Setup SAP HANA System Replication between both sites:

- Copy keys.
- Backup primary database (**SYSTEMDB** and tenant).
- Stop HANA on secondary site.
- Register secondary HANA site to primary HANA site.
- Start HANA on secondary site.

The HANA database installation can also be done using the **hdblcm** command in batch mode. It is possible to use the config file template, which is used as an answer file for a complete automatic installation.

In this solution, the SAP database is installed over the batch mode with the integration of additional hosts that perform an automatic deployment over the SAP Host Agent for each datacenter. A temporary password file is generated, which includes all of the necessary deployment passwords. Based on this file, a command-based batch mode installation starts.

For batch mode installation, the following parameters must be changed:

- SID
- System number
- Hostname of the installation instance (**hostname**)

- All hostnames and roles (**addhosts**)
- System type (**system_usage**)
- Home directory of **<sid>adm** user
- **userid** from user **sapadm**
- **groupid** from **sapsys**

Most of the parameters are provided by SAP.

Procedure

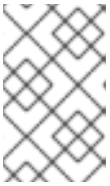
1. Login as user root on one SAP HANA node in each datacenter to start the SAP HANA Scale-Out installation.
2. In this solution, the following command is executed on one node in each datacenter:

```
[root:~]# INSTALLDIR=/install/51053381/DATA_UNITS HDB_SERVER_LINUX_X86_64/
[root:~]# cd $INSTALLDIR
[root:~]# ./hdblcm --dump_configfile_template=/tmp/templateFile
```

IMPORTANT

The correct addhosts parameter must be used. This must not include the installation node.

3. Change the passwords in **/tmp/templateFile.xml**:



NOTE

The **internal_network** parameter is for the internal scale-out communication network. This prefills the SAP HANA configuration file **global.ini** with the correct configuration during the installation process.

Datacenter 1 example:

```
[root:~]# cat /tmp/templateFile.xml | ./hdblcm \ --batch \ --sid=RH1 \ --number=10 \ --
action=install \ --hostname=dc1hana01 \ --
addhosts=dc1hana02:role=worker,dc1hana03:role=worker,dc1hana04:role =standby \ --
install_hostagent \ --system_usage=test \ --sapmnt=/hana/shared \ --datapath=/hana/data \ --
logpath=/hana/log \ --root_user=root \ --workergroup=default \ --home=/usr/sap/RH1/home \ -
-userid=79 \ --shell=/bin/bash \ --groupid=79 \ --read_password_from_stdin=xml \ --
internal_network=192.168.101.0/24 \ --remote_execution=saphostagent
```

Datacenter 2 example:

```
[root:~]# cat /tmp/templateFile.xml | ./hdblcm \ --batch \ --sid=RH1 \ --number=10 \ --
action=install \ --hostname=dc2hana01 \ --
addhosts=dc2hana02:role=worker,dc2hana03:role=worker,dc2hana04:role =standby \ --
install_hostagent \ --system_usage=test \ --sapmnt=/hana/shared \ --datapath=/hana/data \ --
```

```
logpath=/hana/log \ --root_user=root \ --workergroup=default \ --home=/usr/sap/RH1/home \ --userid=79 \ --shell=/bin/bash \ --groupid=79 \ --read_password_from_stdin=xml \ --internal_network=192.168.101.0/24 \ --remote_execution=saphostagent
```

4. Verify that everything is working on one host per datacenter after the installation process is complete:

```
[root:~]# su - rh1adm /usr/sap/hostctrl/exe/sapcontrol -nr 10 -function GetSystemInstanceList
10.04.2019 08:38:21
GetSystemInstanceList OK
hostname, instanceNr, httpPort, httpsPort, startPriority, features, dispstatus
dc1hana01,10,51013,51014,0.3,HDB|HDB_WORKER, GREEN
dc1hana03,10,51013,51014,0.3,HDB|HDB_STANDBY, GREEN
dc1hana02,10,51013,51014,0.3,HDB|HDB_WORKER, GREEN
dc1hana04,10,51013,51014,0.3,HDB|HDB_WORKER, GREEN

rh1adm@dc1hana01:/usr/sap/RH1/HDB10> HDBSettings.sh landscapeHostConfiguration.py
| Host | Host | Host | Failover | Remove | Storage |
Storage | Failover | Failover | NameServer | NameServer |
IndexServer | IndexServer | Host | Host | Worker | Worker |
| | Active | Status | Status | Status | Config |
Actual | Config | Actual | Config | Actual | Config
| Actual | Config | Actual | Config | Actual |
| | | | | | Partition |
Partition | Group | Group | Role | Role | Role
| Role | Roles | Roles | Groups | Groups |
| --------- | ------ | ------ | -------- | ------ | --------- |
--------- | -------- | -------- | ---------- | ---------- |
---------- | ---------- | ------- | ------- | ------- | ------- |
| dc1hana01 | yes | ok | | | 1 |
1 | default | default | master 1 | master | worker |
master | worker | worker | default | default |
| dc1hana02 | yes | ok | | | 2 |
2 | default | default | master 3 | slave | worker |
slave | worker | worker | default | default |
| dc1hana03 | yes | ok | | | 2 |
2 | default | default | master 3 | slave | worker |
slave | worker | worker | default | default |
| dc1hana04 | yes | ignore | | | 0 |
0 | default | default | master 2 | slave | standby |
standby | standby | standby | default | - |

rh1adm@dc1hana01: HDB info
USER PID PPID %CPU VSZ RSS COMMAND
rh1adm 31321 31320 0.0 116200 2824 -bash
rh1adm 32254 31321 0.0 113304 1680 \_ /bin/sh
/usr/sap/RH1/HDB10/HDB info
rh1adm 32286 32254 0.0 155356 1868 \_ ps fx
-U rh1adm -o user:8,pid:8,ppid:8,pcpu:5,vsz:10,rss:10,args
rh1adm 27853 1 0.0 23916 1780 sapstart
pf=/hana/shared/RH1/profile/RH1_HDB10_dc1hana01
rh1adm 27863 27853 0.0 262272 32368 \_
/usr/sap/RH1/HDB10/dc1hana01/trace/hdb.sapRH1_HDB10 -d -nw -f
/usr/sap/RH1/HDB10/dc1hana01/daemon.ini
pf=/usr/sap/RH1/SYS/profile/RH1_HDB10_dc1hana01
rh1adm 27879 27863 53.0 9919108 6193868 \_
```

```
hdbnameserver
rh1adm 28186 27863 0.7 1860416 268304 \_
hdbcompileserver
rh1adm 28188 27863 65.8 3481068 1834440 \_
hdbpreprocessor
rh1adm 28228 27863 48.2 9431440 6481212 \_
hdbindexserver -port 31003
rh1adm 28231 27863 2.1 3064008 930796 \_
hdbxsengine -port 31007
rh1adm 28764 27863 1.1 2162344 302344 \_
hdbwebdispatcher
rh1adm 27763 1 0.2 502424 23376
/usr/sap/RH1/HDB10/exe/sapstartsrvpf=/hana/shared/RH1/profile/RH1_HDB10_dc1hana01 -
D -u rh1adm
```

3.2.5. Configuring SAP HANA System Replication

Configuring SAP HANA System Replication is done after both scale-out environments are installed. The configuration steps are:

- Backup the primary database.
- Enable system replication on the primary database.
- Stop secondary database.
- Copy database keys.
- Register the secondary database.
- Start secondary database.
- Verify system replication.

This solution provides high-level information about each step.

3.2.5.1. Backing up the primary database

Backing up the primary database is required for SAP HANA System Replication. Without it, you cannot bring SAP HANA into a system replication configuration.

NOTE

- This solution provides a simple example. In a production environment, you must take into account your backup infrastructure and setup.
- It is very important that you include "/" in the SQL command; for example, **/hana/shared/backup/**. If you do not, then you need write access to the directory, as SAP HANA will not use the directory but instead will create files named **PATH_databackup***.

```
# Do this as root
[root@dc1hana01]# mkdir -p /hana/shared/backup/
[root@dc1hana01]# chown rh1adm /hana/shared/backup/
[root@dc1hana01]# su - rh1adm
```

```
[rh1adm@dc1hana01]% hdbsql -i 10 -u SYSTEM -d SYSTEMDB "BACKUP DATA USING FILE
('/hana/shared/backup/')"
[rh1adm@dc1hana01]% hdbsql -i 10 -u SYSTEM -d RH1 "BACKUP DATA USING FILE
('/hana/shared/backup/')"
```

3.2.5.2. Enable HANA System Replication

After creating the backup functionality on your datacenter, you can start to configure system replication. The first datacenter starts with the configuration as the source site.

1. Enable system replication on the first datacenter (DC1) on one host of the scale-out system.

   ```
   [root@dc1hana01]# su - rh1adm
   [rh1adm@dc1hana01]% hdbnsutil -sr_enable --name=DC1
   nameserver is active, proceeding …
   successfully enabled system as system replication source site done.
   ```

   After the first datacenter is enabled for system replication, the second datacenter must be registered to the first datacenter. You must copy two keys from the enabled source system to the second datacenter. This must be done when the database is stopped.

2. Copy the key and key data file from the primary site to the secondary site. This is done on only one node in each datacenter. This file is shared over the **/hana/shared** directory in the separated scale-out environments. For more information, see SAP Note 2369981 - Required configuration steps for authentication with HANA System Replication.
   Start this command on one node in Datacenter 1 (DC1):

   ```
   [root@dc1hana01]# scp -rp /usr/sap/RH1/SYS/global/security/rsecssfs/data/SSFS_RH1.DAT
   root@dc2hana01:/usr/sap/RH1/SYS/global/security/rsecssfs/data/SSFS_RH 1.DAT
   [root@dc1hana01]# scp -rp /usr/sap/RH1/SYS/global/security/rsecssfs/key/SSFS_RH1.KEY
   root@dc2hana01:/usr/sap/RH1/SYS/global/security/rsecssfs/key/SSFS_RH1 .KEY
   ```

3. You can register the second datacenter (secondary SAP HANA instance) to the primary SAP HANA instance, after copying both keys to the secondary site. This has to be done on a node from Datacenter 2 (DC2) as `**user <sid>adm**.

   NOTE

   Up to now, two modes for the replication type are available:

   - delta_datashipping
   - logreplay

   The replication mode should be either **sync** or **syncmem**. The "classic" operation mode is **delta_datashipping**. The preferred mode for HA is **logreplay**. Using the operation mode **logreplay** makes your secondary site in SAP HANA System Replication a hot standby system. For more information, see the SAP HANA System Replication .

4. With the preferred operation mode, system replication is configured on the DC2 node as the **<sid>adm** user:

   ```
   [root@dc1hana01]# su - rh1adm
   ```

```
[rh1adm@dc1hana01]% hdbnsutil -sr_register --name=DC2 \ --remoteHost=dc1hana03 --
remoteInstance=10 \ --replicationMode=sync --operationMode=logreplay \ --online

# Start System

[rh1adm@dc1hana01]% /usr/sap/hostctrl/exe/sapcontrol -nr 10 -function StartSystem
```

After the system starts, run the following commands to verify that everything works as expected. When the HANA Scale-Out environment is running correctly, **dispstatus** must show **GREEN** for all nodes in the output of the **GetSystemInstanceList** function of **sapcontrol** (this may take several minutes after initial startup). Also, the landscape host configuration must be in the **OK** state.

```
GetInstanceList: rh1adm@dc2hana01:/usr/sap/RH1/HDB10> /usr/sap/hostctrl/exe/sapcontrol
-nr 10 -function GetSystemInstanceList
01.04.2019 14:17:28
GetSystemInstanceList
OK
hostname, instanceNr, httpPort, httpsPort, startPriority, features, dispstatus
dc2hana02, 10, 51013, 51014, 0.3, HDB|HDB_WORKER, GREEN
dc2hana01, 10, 51013, 51014, 0.3, HDB|HDB_WORKER, GREEN
dc2hana04, 10, 51013, 51014, 0.3, HDB|HDB_STANDBY, GREEN
dc2hana03, 10, 51013, 51014, 0.3, HDB|HDB_WORKER, GREEN

Check landscapeHostConfiguration: rh1adm@dc2hana01:/usr/sap/RH1/HDB10>
HDBSettings.sh landscapeHostConfiguration.py
Storage | Failover | Failover | NameServer | NameServer | IndexServer | IndexServer | Host |
Host | Worker | Worker |
|  | Active | Status | Status | Status | Config | Actual | Config | Actual | Config  | Actual | Config
| Actual | Config | Actual | Config | Actual |
| | | |  |  | Partition | Partition | Group | Group  | Role  | Role | Role
| Role | Roles | Roles | Groups | Groups |
| | | | | | |
| | | | |
| | | | | |
| dc2hana01 | yes | ok | | | 1 |
| default | default | master 1 | master | worker | master | worker | worker | default | default |
| dc2hana02 | yes | ok | | | 2 |
| default | default | slave | slave | worker | slave | worker | worker | default | default |
| dc2hana03 | yes | ok | | | 3 |
| default | default | master 3 | slave | worker | slave | worker | worker | default | default |
| dc2hana04 | yes | ignore | | | 0 |

0 | default
| default | master 2 | slave
|
standby
|
standby
| standby | standby | default | -
|
overall host status: ok
```

5. On the Datacenter 1 site, the **dispstatus** must show **GREEN** for all nodes in the output of the **GetSystemInstanceList** function of **sapcontrol** and the landscape host configuration must be in the **OK** state.

```
rh1adm@dc1hana01: /usr/sap/hostctrl/exe/sapcontrol -nr 10 -function
GetSystemInstanceList rh1adm@dc1hana01:/hana/shared/backup>
/usr/sap/hostctrl/exe/sapcontrol -nr 10 -function
GetSystemInstanceList
Red Hat Enterprise Linux HA Solution for SAP HANA Scale Out and System Replication
Page 55
26.03.2019 12:41:13
GetSystemInstanceList
OK
hostname, instanceNr, httpPort, httpsPort, startPriority, features,
dispstatus
dc1hana01, 10, 51013, 51014, 0.3, HDB|HDB_WORKER, GREEN
dc1hana02, 10, 51013, 51014, 0.3, HDB|HDB_WORKER, GREEN
dc1hana03, 10, 51013, 51014, 0.3, HDB|HDB_WORKER, GREEN
dc1hana04, 10, 51013, 51014, 0.3, HDB|HDB_STANDBY, GREEN

rh1adm@dc1hana01:/usr/sap/RH1/HDB10> HDBSettings.sh landscapeHostConfiguration.py
| Host | Host | Host | Failover | Remove | Storage |
Storage | Failover | Failover | NameServer | NameServer |
IndexServer | IndexServer | Host | Host | Worker | Worker |
| | Active | Status | Status | Status | Config |
Actual | Config | Actual | Config | Actual | Config
| Actual | Config | Actual | Config | Actual |
| | | | | | Partition |
Partition | Group | Group | Role | Role | Role
| Role | Roles | Roles | Groups | Groups |
| --------- | ------ | ------ | -------- | ------ | --------- |
--------- | -------- | -------- | ---------- | ---------- |
---------- | ---------- | ------- | ------- | ------- | ------- |
| dc1hana01 | yes | ok | | | 1 |
1 | default | default | master 1 | master | worker |
master | worker | worker | default | default |
| dc1hana02 | yes | ok | | | 2 |
2 | default | default | master 2 | slave | worker |
slave | worker | worker | default | default |
| dc1hana03 | yes | ok | | | 3 |
3 | default | default | slave | slave | worker |
slave | worker | worker | default | default |
| dc1hana04 | yes | ignore | | | 0 |
0 | default | default | master 3 | slave | standby |
Red Hat Enterprise Linux HA Solution for SAP HANA Scale Out and System Replication
Page 56
standby | standby | standby | default | - |
overall host status: ok
rh1adm@dc1hana01:/usr/sap/RH1/HDB10>

# Show Systemreplication state rh1adm@dc1hana01:/usr/sap/RH1/HDB10> HDBSettings.sh
systemReplicationStatus.py
| Database | Host | Port | Service Name | Volume ID | Site ID |
Site Name | Secondary | Secondary | Secondary | Secondary | Secondary
| Replication | Replication | Replication |
| | | | | | |
```

```
| Host | Port | Site ID | Site Name | Active Status |
Mode | Status | Status Details |
| -------- | --------- | ----- | ------------ | --------- | ------- |
--------- | --------- | --------- | --------- | --------- |
-------------- | ------------ | ------------ | --------------- |
| SYSTEMDB | dc1hana01 | 31001 | nameserver | 1 | 1 |
DC1 | dc2hana01 | 31001 | 2 | DC2 | YES
| SYNC | ACTIVE | |
| RH1 | dc1hana01 | 31007 | xsengine | 2 | 1 |
DC1 | dc2hana01 | 31007 | 2 | DC2 | YES
| SYNC | ACTIVE | |
| RH1 | dc1hana01 | 31003 | indexserver | 3 | 1 |
DC1 | dc2hana01 | 31003 | 2 | DC2 | YES
| SYNC | ACTIVE | |
| RH1 | dc1hana03 | 31003 | indexserver | 5 | 1 |
DC1 | dc2hana03 | 31003 | 2 | DC2 | YES
| SYNC | ACTIVE | |
| RH1 | dc1hana02 | 31003 | indexserver | 4 | 1 |
DC1 | dc2hana02 | 31003 | 2 | DC2 | YES
| SYNC | ACTIVE | |
status system replication site "2": ACTIVE
overall system replication status: ACTIVE
Local System Replication State
Red Hat Enterprise Linux HA Solution for SAP HANA Scale Out and System Replication
Page 57
~~~~~~~~~~
mode: PRIMARY
site id: 1
site name: DC1
rh1adm@dc1hana01:/usr/sap/RH1/HDB10>
```

6. Check if HANA System Replication is active.

```
rh1adm@dc1hana01:/usr/sap/RH1/HDB10> HDBSettings.sh systemReplicationStatus.py
| Database | Host | Port | Service Name | Volume ID | Site ID |
Site Name | Secondary | Secondary | Secondary | Secondary | Secondary
| Replication | Replication | Replication |
| | | | | | |
| Host | Port | Site ID | Site Name | Active Status |
Mode | Status | Status Details |
| -------- | --------- | ----- | ------------ | --------- | ------- |
--------- | --------- | --------- | --------- | --------- |
-------------- | ------------ | ------------ | --------------- |
| SYSTEMDB | dc1hana01 | 31001 | nameserver | 1 | 1 |
DC1 | dc2hana01 | 31001 | 2 | DC2 | YES
| SYNC | ACTIVE | |
| RH1 | dc1hana01 | 31007 | xsengine | 2 | 1 |
DC1 | dc2hana01 | 31007 | 2 | DC2 | YES
| SYNC | ACTIVE | |
| RH1 | dc1hana01 | 31003 | indexserver | 3 | 1 |
DC1 | dc2hana01 | 31003 | 2 | DC2 | YES
| SYNC | ACTIVE | |
| RH1 | dc1hana03 | 31003 | indexserver | 5 | 1 |
DC1 | dc2hana03 | 31003 | 2 | DC2 | YES
| SYNC | ACTIVE | |
| RH1 | dc1hana02 | 31003 | indexserver | 4 | 1 |
```

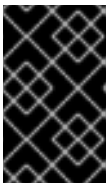
```
DC1 | dc2hana02 | 31003 | 2 | DC2 | YES
| SYNC | ACTIVE | |

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

Local System Replication State
~~~~~~~~~~
mode: PRIMARY
site id: 1
site name: DC1
rh1adm@dc1hana01:/usr/sap/RH1/HDB10>
```

NOTE

If this configuration is implemented in a production environment, it is recommended that you change the network communication in the **global.ini** file. This action limits the communication to a specified adapter to the system replication network. For more information, see the Network Configuration for SAP HANA system replication.



IMPORTANT

It is necessary to manually test the complete SAP HANA Scale-Out System Replication environment and verify that all SAP HANA features are working. For more information, see the SAP HANA System Replication .

3.3. CONFIGURING PACEMAKER

When the HANA Scale-Out environment is configured and HANA system replication is working as expected, you can configure the HA cluster to manage the HANA Scale-Out System Replication environment using the RHEL HA Add-On.

An additional quorum instance is necessary to prevent a Pacemaker split-brain configuration. In this example, we add a node. This node, referred to in this solution as **majoritymaker**, is needed for an odd number of cluster nodes for a working configuration. This is an additional minimalist host that only requires the Pacemaker and public network. While on this node, no SAP HANA database is installed, and storage configuration is obsolete.

Prerequisites

- You have installed the **saphostagent** and checked if /usr/sap/hostcontrol/exe/sapcontrol exists. For more information, check 1031096 - Installing Package SAPHOSTAGENT .
- You have verified that the RHEL High Availability repository is configured in the system. You cannot install Pacemaker without this configuration.
- You have logged in as root to all systems.
- You have verified that all cluster nodes are registered and have the required repositories enabled to install the packages for the cluster, as described in the Registering your RHEL system and enabling repositories section of this document.

```
[root@dc1hana01]# subscription-manager repos --list-enabled
```

```
+----------------------------------------------------------+
 Available Repositories in /etc/yum.repos.d/redhat.repo
+----------------------------------------------------------+
Repo ID:   rhel-8-for-x86_64-baseos-e4s-rpms
Repo Name: Red Hat Enterprise Linux 8 for x86_64 - BaseOS - Update Services for SAP
Solutions (RPMs)
Repo URL:  <Your repo URL>
Enabled:   1

Repo ID:   rhel-8-for-x86_64-sap-solutions-e4s-rpms
Repo Name: Red Hat Enterprise Linux 8 for x86_64 - SAP Solutions - Update Services for
SAP Solutions (RPMs)
Repo URL:  <Your repo URL>
Enabled:   1

Repo ID:   ansible-2.8-for-rhel-8-x86_64-rpms
Repo Name: Red Hat Ansible Engine 2.8 for RHEL 8 x86_64 (RPMs)
Repo URL:  <Your repo URL>
Enabled:   1

Repo ID:   rhel-8-for-x86_64-highavailability-e4s-rpms
Repo Name: Red Hat Enterprise Linux 8 for x86_64 - High Availability - Update Services for
SAP Solutions (RPMs)
Repo URL:  <Your repo URL>
Enabled:   1

Repo ID:   rhel-8-for-x86_64-appstream-e4s-rpms
Repo Name: Red Hat Enterprise Linux 8 for x86_64 - AppStream - Update Services for SAP
Solutions (RPMs)
Repo URL:  <Your repo URL>
Enabled:   1


yum repolist
Updating Subscription Management repositories.
repo id                                                repo name
advanced-virt-for-rhel-8-x86_64-rpms                        Advanced Virtualization for
RHEL 8 x86_64 (RPMs)
ansible-2.8-for-rhel-8-x86_64-rpms                        Red Hat Ansible Engine 2.8 for
RHEL 8 x86_64 (RPMs)
rhel-8-for-x86_64-appstream-e4s-rpms                        Red Hat Enterprise Linux 8 for
x86_64 - AppStream - Update Services for SAP Solutions (RPMs)
rhel-8-for-x86_64-baseos-e4s-rpms                        Red Hat Enterprise Linux 8 for
x86_64 - BaseOS - Update Services for SAP Solutions (RPMs)
rhel-8-for-x86_64-highavailability-e4s-rpms                        Red Hat Enterprise Linux 8 for
x86_64 - High Availability - Update Services for SAP Solutions (RPMs)
rhel-8-for-x86_64-sap-netweaver-e4s-rpms                        Red Hat Enterprise Linux 8
for x86_64 - SAP NetWeaver - Update Services for SAP Solutions (RPMs)
rhel-8-for-x86_64-sap-solutions-e4s-rpms                        Red Hat Enterprise Linux 8 for
x86_64 - SAP Solutions - Update Services for SAP Solutions (RPMs)
```

Procedure

1. Configure the cluster. For more information, see Configuring and managing high availability clusters.

2. On each node in the cluster, including the **majoritymaker**, install the Red Hat High Availability Add-On software packages along with all available fence agents from the High Availability channel:

```
[root]# yum -y install pcs pacemaker fence-agents
```

Alternatively, you can also install only specific fence-agents:

```
[root]# yum install fence-agents-sbd fence-agents-ipmilan
```

3. Execute the following commands to enable the ports that are required by the Red Hat High Availability Add-On, if you are running the **firewalled** daemon:

```
[root]# firewall-cmd --permanent --add-service=high-availability
[root]# firewall-cmd --add-service=high-availability
```

4. After this configuration, set the password for the user **hacluster** on each cluster node.

```
[root]# passwd hacluster

Changing password for user hacluster. New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

5. Start and enable the daemon by issuing the following commands on each node:

```
[root]# systemctl start [root]# pcsd.service systemctl enable pcsd.service
```

6. On only one node, you have to authenticate the **hacluster** user. It is important to include every node in this command, which should be part of the cluster. If you don't specify the password, you are asked for the **hacluster** password, which was defined in the previous step. For RHEL 8.x run the following:

```
[root@dc1hana01]# pcshost auth -u hacluster -p <clusterpassword> dc1hana01 dc1hana02
dc1hana03 dc1hana04 dc2hana01 dc2hana02 dc2hana03 dc2hana04 majoritymaker
Username: hacluster
Password:
majoritymaker: Authorized
dc1hana03: Authorized
dc1hana02: Authorized
dc1hana01: Authorized
dc2hana01: Authorized
dc2hana02: Authorized
dc1hana04: Authorized
dc2hana04: Authorized
dc2hana03: Authorized
```

7. Use the **pcs cluster** setup on the same node to generate and synchronize the **corosync** configuration. The RHEL 8 example also shows, if you are using 2 cluster network

```
[root@dc1hana01]# pcs cluster setup scale_out_hsr majoritymaker addr=10.10.10.41
addr=192.168.102.100 dc1hana01 addr=10.10.10.21 addr=192.168.102.101 dc1hana02
addr=10.10.10.22 addr=192.168.102.102 dc1hana03 addr=10.10.10.23
```

```
addr=192.168.102.103 dc1hana04 addr=10.10.10.24 addr=192.168.102.104 dc2hana01
addr=10.10.10.31 addr=192.168.102.201 dc2hana02 addr=10.10.10.33
addr=192.168.102.202 dc2hana03 addr=10.10.10.34 addr=192.168.212.203 dc2hana04
addr=10.10.10.10 addr=192.168.102.204

Destroying cluster on nodes: dc1hana01, dc1hana02, dc1hana03,
dc1hana04, dc2hana01, dc2hana02, dc2hana03, dc2hana04,
majoritymaker...
dc1hana01: Stopping Cluster (pacemaker)...
dc1hana04: Stopping Cluster (pacemaker)...
dc1hana03: Stopping Cluster (pacemaker)...
dc2hana04: Stopping Cluster (pacemaker)...
dc2hana01: Stopping Cluster (pacemaker)...
dc2hana03: Stopping Cluster (pacemaker)...
majoritymaker: Stopping Cluster (pacemaker)...
dc2hana02: Stopping Cluster (pacemaker)...
dc1hana02: Stopping Cluster (pacemaker)...
dc2hana01: Successfully destroyed cluster
dc2hana03: Successfully destroyed cluster
dc1hana04: Successfully destroyed cluster
dc1hana03: Successfully destroyed cluster
dc2hana02: Successfully destroyed cluster
dc1hana01: Successfully destroyed cluster
dc1hana02: Successfully destroyed cluster
dc2hana04: Successfully destroyed cluster
majoritymaker: Successfully destroyed cluster
Sending 'pacemaker_remote authkey' to 'dc1hana01', 'dc1hana02',
'dc1hana03', 'dc1hana04', 'dc2hana01', 'dc2hana02', 'dc2hana03',
'dc2hana04', 'majoritymaker'
dc1hana01: successful distribution of the file 'pacemaker_remote
authkey'
dc1hana04: successful distribution of the file 'pacemaker_remote
authkey'
dc1hana03: successful distribution of the file 'pacemaker_remote
authkey'
dc2hana01: successful distribution of the file 'pacemaker_remote
authkey'
dc2hana02: successful distribution of the file 'pacemaker_remote
authkey'
dc2hana03: successful distribution of the file 'pacemaker_remote
authkey'
dc2hana04: successful distribution of the file 'pacemaker_remote
authkey'
majoritymaker: successful distribution of the file 'pacemaker_remote
authkey'
dc1hana02: successful distribution of the file 'pacemaker_remote
authkey'
Sending cluster config files to the nodes...
dc1hana01: Succeeded
dc1hana02: Succeeded
dc1hana03: Succeeded
dc1hana04: Succeeded
dc2hana01: Succeeded
dc2hana02: Succeeded
dc2hana03: Succeeded
dc2hana04: Succeeded
```

```
majoritymaker: Succeeded
Starting cluster on nodes: dc1hana01, dc1hana02, dc1hana03,
dc1hana04, dc2hana01, dc2hana02, dc2hana03, dc2hana04,
majoritymaker...
dc2hana01: Starting Cluster...
dc1hana03: Starting Cluster...
dc1hana01: Starting Cluster...
dc1hana02: Starting Cluster...
dc1hana04: Starting Cluster...
majoritymaker: Starting Cluster...
dc2hana02: Starting Cluster...
dc2hana03: Starting Cluster...
dc2hana04: Starting Cluster...
Synchronizing pcsd certificates on nodes dc1hana01, dc1hana02,
dc1hana03, dc1hana04, dc2hana01, dc2hana02, dc2hana03, dc2hana04,
majoritymaker...
majoritymaker: Success
dc1hana03: Success
dc1hana02: Success
dc1hana01: Success
dc2hana01: Success
dc2hana02: Success
dc2hana03: Success
dc2hana04: Success
dc1hana04: Success
Restarting pcsd on the nodes in order to reload the certificates...
dc1hana04: Success
dc1hana03: Success
dc2hana03: Success
majoritymaker: Success
dc2hana04: Success
dc1hana02: Success
dc1hana01: Success
dc2hana01: Success
dc2hana02: Success
```

8. Enable the services on every node with the following cluster command:

```
[root@dc1hana01]# pcs cluster enable --all
dc1hana01: Cluster Enabled
dc1hana02: Cluster Enabled
dc1hana03: Cluster Enabled
dc1hana04: Cluster Enabled
dc2hana01: Cluster Enabled
dc2hana02: Cluster Enabled
dc2hana03: Cluster Enabled
dc2hana04: Cluster Enabled
majoritymaker: Cluster Enabled
```

Completing all steps results in a configured cluster and nodes. The first step in configuring the resource agents is to configure the fencing method with STONITH, which reboots nodes that are no longer accessible. This STONITH configuration is required for a supported environment.

9. Use the fence agent that is appropriate for your hardware or virtualization environment to configure STONITH for the environment. Below is a generic example of configuring a fence device for STONITH:

```
[root@dc1hana01]# pcs stonith create <stonith id> <fence_agent> ipaddr=<fence device>
login=<login> passwd=<passwd>
```

NOTE

Configuration for each device is different, and configuring STONITH is a requirement for this environment. If you need assistance, please contact Red Hat Support for direct assistance. For more information, refer to Support Policies for RHEL High Availability Clusters - General Requirements for Fencing/STONITH and Fencing Configuration.

After configuration, the cluster status should look like the following output. This is an example of a fencing device of a Red Hat Enterprise virtualization environment:

```
[root@dc1hana01]# pcs status
Cluster name: hanascaleoutsr
Stack: corosync
Current DC: dc2hana01 (version 1.1.18-11.el7_5.4-2b07d5c5a9) -
partition with quorum
Last updated: Tue Mar 26 13:03:01 2019
Last change: Tue Mar 26 13:02:54 2019 by root via cibadmin on
dc1hana01
9 nodes configured
1 resource configured
Online: [ dc1hana01 dc1hana02 dc1hana03 dc1hana04 dc2hana01 dc2hana02
dc2hana03 dc2hana04 majoritymaker ]
Full list of resources:
fencing (stonith:fence_rhevm): Started dc1hana01
Daemon Status:
corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled
```

3.3.1. Installing SAP HANA resource agents for Scale-Out systems

When configuring the resource agents, the **resource-agent-sap-hana-scaleout** package was installed on every system, including the **majoritymaker**:

```
[root@dc1hana01]# yum install resource-agents-sap-hana-scaleout
```

Verify that the correct repository is attached. **yum repolist** should contain:

```
root# yum repolist
"rhel-x86_64-server-sap-hana-<version>" RHEL Server SAP HANA (v. <version> for 64-bit
<architecture>).
```

3.3.2. Enabling the srConnectionChanged() hook on all SAP HANA instances

As documented in SAP's Implementing a HA/DR Provider , recent versions of SAP HANA provide "hooks" that allow SAP HANA to send out notifications for certain events. The **srConnectionChanged()** hook can be used to improve the ability of the cluster to detect when a change in the status of the SAP HANA System Replication occurs that requires the cluster to take action and to avoid data loss/data

corruption by preventing accidental takeovers to be triggered in situations where this should be avoided. You must enable the hook before proceeding with the cluster setup, when using SAP HANA 2.0 SPS06 or later and a version of the **resource-agents-sap-hana-scaleout** package that provides the components for supporting the **srConnectionChanged()** hook.

Procedure

1. Install the hook on one node in each datacenter on a shared device. For more information, see Implementing a HA/DR Provider .

2. Create a directory in the hana shared folder to configure the hooks. This is configured to create additional data from the SAP HANA database. To enable it, you must stop the system and add two additional configuration parameters to the **global.ini** file. In this solution, the following example shows the configuration of **ha_dr_provider_SAPHanaSr** and **trace**.

```
[root@dc1hana01]# su - rh1adm
[rh1adm@dc1hana01]% sapcontrol -nr 10 -function StopSystem *[rh1adm@dc1hana01]%
cat <<EOF >> /hana/shared/RH1/global/hdb/custom/config/global.ini

[ha_dr_provider_SAPHanaSR]
provider = SAPHanaSR
path = /usr/share/SAPHanaSR-ScaleOut
execution_order = 1

[trace]
ha_dr_saphanasr = info
EOF
```

3. On each cluster node, create the file **/etc/sudoers.d/20-saphana** by running **sudo visudo /etc/sudoers.d/20-saphana** and add the contents below to allow the hook script to update the node attributes when the **srConnectionChanged()** hook is called:

```
rh1adm ALL=(ALL) NOPASSWD: /usr/sbin/crm_attribute -n hana_rh1_glob_srHook -v * -t
crm_config -s SAPHanaSR

rh1adm ALL=(ALL) NOPASSWD: /usr/sbin/crm_attribute -n hana_rh1_gsh -v * -l reboot -t
crm_config -s SAPHanaSR Defaults:rh1adm !requiretty
```

For further information on why the defaults setting is needed, see The srHook attribute is set to SFAIL in a Pacemaker cluster managing SAP HANA system replication, even though replication is in a healthy state.

4. Start the SAP HANA database after the successful integration.

```
# Execute the following commands on one HANA node in every datacenter
[root]# su - rh1adm
[rh1adm]% sapcontrol -nr 10 -function StartSystem
```

5. Verify that the hook script is working as expected. Perform an action to trigger the hook, such as stopping the HANA instance. Then, use the method given below to check if the hook logged anything:

```
[rh1adm@dc1hana01]% cdtrace
[rh1adm@dc1hana01]% awk '/ha_dr_SAPHanaSR.*crm_attribute/ \
{ printf "%s %s %s %s\n",$2,$3,$5,$16 }' nameserver_* 2018-05-04
```

```
12:34:04.476445 ha_dr_SAPHanaSR SFAIL
2018-05-04 12:53:06.316973 ha_dr_SAPHanaSR SOK
```

For more information on how to verify that the SAP HANA hook is working, see Monitoring with M_HA_DR_PROVIDERS.

3.3.3. Configuring Pacemaker resources

You have to create two resource agents, **SAPHanaTopology** and **SAPHanaController**, that control the HANA and Pacemaker environment, for the Pacemaker configuration process. Additionally, you need to configure a virtual IP address in Pacemaker for the connectivity of the end-user and the SAP application server. Based on the actions performed, two dependencies are added to ensure that the resource agents are executed in the correct order and that the virtual IP address is mapped to the right host.

Prerequisites

- You have set the cluster **maintenance-mode** to avoid unwanted effects during configuration:

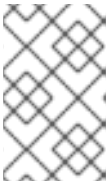
```
[root@dc1hana01]# pcs property set maintenance-mode=true
```

3.3.3.1. Configuring the SAPHanaTopology resource

1. The **SAPHanaTopology** resource agent gathers the status and configuration of SAP HANA System Replication on each node. In addition, it starts and monitors the local SAP HostAgent which is required to start, stop, and monitor the SAP HANA instances. The resource agent has the following attributes that depend on the installed SAP HANA environment:

Attribute Name	Required?	Default value	Description
SID	yes	null	The SAP System Identifier (SID) of the SAP HANA installation (must be identical for all nodes). Example: RH2
InstanceNumber	yes	null	The Instance Number of the SAP HANA installation (must be identical for all nodes). Example: 02

In this solution, the SID is set to RH1 and the Instance Number is set to 10.



NOTE

The timeout and monitor parameters are recommended for the first deployment, and they can be changed while testing the environment. There are several dependencies, like the size and the number of nodes in the environment.

2. Execute the following command for RHEL 8.x as root on one host in the whole cluster:

```
[root@dc1hana01]# pcs resource create rsc_SAPHanaTopology_RH1_HDB10
SAPHanaTopology SID=RH1 InstanceNumber=10 op methods interval=0s timeout=5 op
monitor interval=10 timeout=600 clone clone-max=6 clone-node-max=1 interleave=true --
disabled
```

When the resource is created in Pacemaker, it is then cloned.

NOTE

The **clone-node-max** parameter defines how many copies of the resource agent can be started on a single node. Interleave means that if this clone depends on another clone using an ordering constraint, it is allowed to start after the local instance of the other clone starts, rather than waiting for all instances of the other clone to start. The **clone-max** parameter defines how many clones could be started; if you have, for example, the minimum configuration of 2 nodes per site, you should use **clone-max=4** for **SAPHanaController** and **SAPHanaTopology**. At 3 nodes per site (without counting the standby node), you should use 6.

3. You can view the collected information stored in the form of node attributes once the resource starts using the command:

```
root# pcs status --full
```

3.3.3.2. Configuring the SAPHanaController resource

When the configuration process for the **SAPHanaTopology** resource agent is complete, the **SAPHanaController** resource agent can be configured. While the SAP Hana Topology resource agent collects only data, the **SAPHanaTopology** resource agent controls the SAP environment based on the data previously collected. As shown in the following table, five important configuration parameters define the cluster functionality:

Attribute Name	Required?	Default value	Description
SID	yes	null	The SAP System Identifier (SID) of the SAP HANA installation (must be identical for all nodes). Example: RH2
InstanceNumber	yes	null	The Instance Number of the SAP HANA installation (must be identical for all nodes). Example: 02

PREFER_SITE_TAKEOVER	no	null	Should resource agent prefer to switch over to the secondary instance instead of restarting primary locally? true: prefer takeover to the secondary site; false: prefer restart locally; never: under no circumstances initiate a takeover to the other node.
AUTOMATED_REGISTER	no	false	If a takeover event has occurred, and the DUPLICATE_PRIMARY_TIMEOUT has expired, should the former primary instance be registered as secondary? ("false": no, manual intervention will be needed; "true": yes, the former primary will be registered by resource agent as secondary) [1].
DUPLICATE_PRIMARY_TIMEOUT	no	7200	The time difference (in seconds) needed between two primary timestamps, if a dual-primary situation occurs. If the time difference is less than the time gap, the cluster will hold one or both instances in a "WAITING" status. This is to give the system admin a chance to react to a takeover. After the time difference has passed, if AUTOMATED_REGISTER is set to true, the failed former primary will be registered as secondary. After the registration to the new primary, all data on the former primary will be overwritten by the system replication.

[1] - As a best practice for testing and Proof of Concept (PoC) environments, it is recommended that you leave **AUTOMATED_REGISTER** at its default value (**AUTOMATED_REGISTER="false"**) to prevent a failed primary instance automatically registering as a secondary instance. After testing, if the failover scenarios work as expected, particularly in a production environment, it is recommended that you set **AUTOMATED_REGISTER="true"** so that after a takeover, system replication will resume in a timely manner, avoiding disruption. When **AUTOMATED_REGISTER="false"** in case of a failure on the primary node, you must manually register it as the secondary HANA system replication node.

The following command is an example for RHEL 8.x of how to create the **SAPHanaController** promotable resource. The example is based on the parameters: **SID RH1**, **InstanceNumber 10**, the values **true** for **Prefer Site Takeover** and **Automated_REGISTER**, and **Duplicate Primary Timeout** of **7200**:

```
[root@dc1hana01]# pcs resource create rsc_SAPHana_RH1_HDB10 SAPHanaController SID=RH1
InstanceNumber=10 PREFER_SITE_TAKEOVER=true DUPLICATE_PRIMARY_TIMEOUT=7200
AUTOMATED_REGISTER=true op demote interval=0s timeout=320 op methods interval=0s
timeout=5 op monitor interval=59 role="Promoted" timeout=700 op monitor interval=61
role="Unpromoted" timeout=700 op promote interval=0 timeout=3600 op start interval=0
timeout=3600 op stop interval=0 timeout=3600 promotable clone-max=6 promoted-node-max=1
interleave=true --disabled
```

NOTE

For **clone-max**, use twice the number of **HDB_WORKERs** listed in the command:

```
/usr/sap/hostctrl/exe/sapcontrol -nr 10 -function GetSystemInstanceList

GetSystemInstanceList
OK
hostname, instanceNr, httpPort, httpsPort, startPriority, features, dispstatus
dc1hana01,10,51013,51014,0.3,HDB|HDB_WORKER,GREEN
dc1hana02,10,51013,51014,0.3,HDB|HDB_WORKER,GREEN
dc1hana03,10,51013,51014,0.3,HDB|HDB_WORKER,GREEN
dc1hana04,10,51013,51014,0.3,HDB|HDB_STANDBY, GREEN
```

In this solution, after the creation of the **SAPHanaController**, the resource is defined as a promotable resource with the following command: (**SID** is **RH1** and **InstanceNumber** is **10**).

For more information, see Multi-State Resources: Resources That Have Multiple Modes .

3.3.3.3. Configuring the resource to manage the virtual IP address

The cluster needs to include a resource to manage the virtual IP address that is used by clients to reach the master nameserver of the primary SAP HANA Scale-Out site.

The following command is an example of how to create an **IPaddr2** resource with the virtual IP 10.0.0.250:

```
[root@dc1hana01]# pcs resource create rsc_ip_SAPHana_RH1_HDB10 ocf:heartbeat:IPaddr2
ip=10.0.0.250 op monitor interval="10s" timeout="20s"
```

3.3.4. Creating constraints

For correct operation, verify that **SAPHanaTopology** resources start before **SAPHanaController** resources start, and also that the virtual IP address is present on the node where the promoted resource of **SAPHanaController** runs. Use this procedure to create the four required constraints.

Procedure: Starting SAPHanaTopology before SAPHana

The following command is an example of how to create the constraint that mandates the start order of the resources.

1. Create the constraint:

```
[root@dc1hana01]# pcs constraint order start rsc_SAPHanaTopology_RH1_HDB10-clone
then start rsc_SAPHana_RH1_HDB10-clone
```

2. Colocate the **IPaddr2** resource with the promoted **SAPHana** resource:

```
[root@dc1hana01]# pcs constraint colocation add rsc_ip_SAPHana_RH1_HDB10 with
promoted rsc_SAPHana_RH1_HDB10-clone
```

3. Avoid the **majoritymaker** to use an active role in the cluster environment:

```
[root@dc1hana01]# pcs constraint location add topology-avoids-majoritymaker
rsc_SAPHanaTopology_RH1_HDB10-clone majoritymaker -INFINITY resource-
discovery=never

[root@dc1hana01]# pcs constraint location add hana-avoids-majoritymaker
rsc_SAPHana_RH1_HDB10-clone majoritymaker -INFINITY resource-discovery=never
```

4. Disable **maintenance-mode**:
   Use **maintenance-mode** to start the resources after setting **maintenance-mode** to false. To avoid activities of pacemaker before all configurations are finished, we have used in the examples above **--disabled**. By default, the resources are started as soon as they are created. With **--disabled**, you can start the resources by using the command:

```
[root@dc1hana01]# pcs resource enable <resource-name>
```

   To leave **maintenance-mode**, please use:

```
[root@dc1hana01]# pcs property set maintenance-mode=false
```

5. Run the following 3 commands to verify that the cluster environment is working correctly:

   - **pcs status** provides an overview of every resource and if they are functioning correctly.
   - **pcs status --full** provides an overview of all resources and additional attribute information of the cluster environment.
   - **SAPHanaSR-showAttr --sid=RH1** provides a readable overview that is based on the attribute information.
     The correct status is displayed a few minutes after deactivating the **maintenance-mode**.

```
[root@dc1hana01]# pcs status
Cluster name: hanascaleoutsr Stack: corosync
Current DC: dc2hana01 (version 1.1.18-11.el7_5.4-2b07d5c5a9) - partition with quorum
```

```
Last updated: Tue Mar 26 14:26:38 2019
Last change: Tue Mar 26 14:25:47 2019 by root via crm_attribute on dc1hana01
9 nodes configured
20 resources configured
Online: [ dc1hana01 dc1hana02 dc1hana03 dc1hana04 dc2hana01 dc2hana02
dc2hana03 dc2hana04 majoritymaker ]
Full list of resources:

fencing (stonith:fence_rhevm): Started dc1hana01
Clone Set: rsc_SAPHanaTopology_RH1_HDB10-clone
[rsc_SAPHanaTopology_RH1_HDB10]
Started: [ dc1hana01 dc1hana02 dc1hana03 dc1hana04 dc2hana01 dc2hana02
dc2hana03 dc2hana04 ]
Stopped: [ majoritymaker ] Clone Set: msl_rsc_SAPHana_RH1_HDB10
[rsc_SAPHana_RH1_HDB10] (promotable):
Promoted: [ dc1hana01 ]
Unpromoted: [ dc1hana02 dc1hana03 dc1hana04 dc2hana01 dc2hana02 dc2hana03
dc2hana04 ]
Stopped: [ majoritymaker ]
rsc_ip_SAPHana_RH1_HDB10 (ocf::heartbeat:IPaddr2): Started dc1hana01
Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled


[root@dc1hana01]# SAPHanaSR-showAttr --sid=RH1
Global prim srHook sync_state
-------------------------------
global DC1 SOK SOK
Sit lpt lss mns srr
---------------------------------
DC1 1553607125 4 dc1hana01 P
DC2 30 4 dc2hana01 S

H clone_state roles score site
-------------------------------------------------------------
1 PROMOTED promoted1 promoted:worker promoted 150 DC1
2 DEMOTED promoted2:slave:worker:slave 110 DC1
3 DEMOTED slave:slave:worker:slave -10000 DC1
4 DEMOTED promoted3:slave:standby:standby 115 DC1
5 DEMOTED promoted2 promoted:worker promoted 100 DC2
6 DEMOTED promoted3:slave:worker:slave 80 DC2
7 DEMOTED slave:slave:worker:slave -12200 DC2
8 DEMOTED promoted1:slave:standby:standby 80 DC2
9 :shtdown:shtdown:shtdown
```

CHAPTER 4. OPTIONAL SETTINGS

4.1. ADDING A SECONDARY VIRTUAL IP ADDRESS RESOURCE FOR ACTIVE/ACTIVE (READ-ENABLED) SETUP

Starting with SAP HANA 2.0 SPS1, SAP allows 'Active/Active (Read Enabled)' setups for SAP HANA System Replication. This allows you to:

- Enable SAP HANA System Replication to support read access on the secondary systems.
- Execute read-intense reporting on the secondary systems to remove this workload from the primary system.
- Reduce the need for bandwidth in continuous operation.

For more information, also check SAP HANA System Replication .

A second virtual IP address is required to allow clients to access the secondary SAP HANA database. In terms of a failure, if the secondary site is not accessible, the second IP will be switched to the primary site to avoid downtime of the read-only access.

The **operationMode** should be set to **logreplay_readaccess**. The second virtual IP and the additional necessary constraints can be configured with the following commands:

```
root# pcs resource create rsc_ip2_SAPHana_RH1_HDB10 ocf:heartbeat:IPaddr2 ip=10.0.0.251 op
monitor interval="10s" timeout="20s
```

4.1.1. Configuring additional constraints

The constraints listed above are strictly recommended. To adjust the behaviour to your environment, additional constraints are required. Examples for those are:

```
root# pcs constraint location rsc_ip_SAPHana_RH1_HDB10 rule score=500 role=master
hana_rh1_roles eq "master1:master:worker:master" and hana_rh1_clone_state eq PROMOTED
```

Move the **IP2** to the primary site in the event that the secondary site goes down:

```
root# pcs constraint location rsc_ip2_SAPHana_RH1_HDB10 rule score=50
id=vip_slave_master_constraint hana_rh1_roles eq 'master1:master:worker:master'
```

```
root# pcs constraint order promote rsc_SAPHana_RH1_HDB10-clone then start
rsc_ip_SAPHana_RH1_HDB10
```

```
root# pcs constraint order start rsc_ip_SAPHana_RH1_HDB10 then start
rsc_ip2_SAPHana_RH1_HDB10
```

```
root# pcs constraint colocation add rsc_ip_SAPHana_RH1_HDB10 with Master
rsc_SAPHana_RH1_HDB10-clone 2000
```

```
root# pcs constraint colocation add rsc_ip2_SAPHana_RH1_HDB10 with Slave
rsc_SAPHana_RH1_HDB10-clone 5
```

Procedure

1. Test the behavior if the cluster is up an running you can run

```
root# watch pcs status
```

2. Stop the secondary HANA instance manually with:

```
sidadm% sapcontrol -nr ${TINSTANCE} -function StopSystem HDB
```

3. After a few seconds the 2nd IP address will be moved to the primary hosts. Then you can manually start the database again with:

```
sidadm% sapcontrol -nr ${TINSTANCE} -function StartSystem HDB
```

4. Restart the cluster, for further usage.

4.2. ADDING FILESYSTEM MONITORING

Pacemaker does not actively monitor mount points unless filesystem resources manage them. In a scale-out environment, the databases can be distributed over different availability zones. Mount points can be available specific to a zone, which then needs to be specified as node attributes. If mounts should only be handled in filesystem resources, then they should be removed from **/etc/fstab**. Mounts are required to run SAP HANA services, hence, before SAP HANA services start, order constraints must ensure that filesystems are mounted. For further information, check How do I configure SAP HANA Scale-Out System Replication in a Pacemaker cluster when the HANA filesystems are on NFS shares?.

4.2.1. Filesystem resource example

An example configuration looks like this:

Listing **pcs node** attribute:

```
[root@dc1hana01]# pcs node attribute
Node Attributes:
 saphdb1: hana_hdb_gra=2.0 hana_hdb_site=DC1 hana_hdb_vhost=sapvirthdb1
 saphdb2: hana_hdb_gra=2.0 hana_hdb_site=DC1 hana_hdb_vhost=sapvirthdb2
 saphdb3: hana_hdb_gra=2.0 hana_hdb_site=DC2 hana_hdb_vhost=sapvirthdb3
 saphdb4: hana_hdb_gra=2.0 hana_hdb_site=DC2 hana_hdb_vhost=sapvirthdb4
```

Please note that **pcs node** attribute and **saphdb1 hana_hdb_site=DC1** attribute names are in lower-case.

Assuming we have the current configuration:

- **SID=RH1**
- **Instance_Number=10**

Node	AZ	Attribute	Value
dc1hana01	DC1	NFS_SHARED_RH1_SITE	DC1

dc1hana02	DC1	NFS_SHARED_RH1_SITE	DC1
dc2hana01	DC2	NFS_SHARED_RH1_SITE	DC2
dc2hana02	DC2	NFS_SHARED_RH1_SITE	DC2

Below is the example to set the node attributes mount points for data and logs which can be handled similarly:

```
[root@dc1hana01]# pcs resource create nfs_hana_shared_dc1 ocf:heartbeat:Filesystem
device=svm-012ab34cd45ef67.fs-0879de29a7fbb752d.fsx.ap-southeast-
2.amazonaws.com:/sap_hana_dc1_log_shared/shared directory=/hana/shared fstype=nfs
options=defaults,suid op monitor interval=60s on-fail=fence timeout=20s OCF_CHECK_LEVEL=20
clone
[root@dc1hana01]# pcs resource create nfs_hana_log_dc1 ocf:heartbeat:Filesystem device=svm-
012ab34cd45ef67.fs-0879de29a7fbb752d.fsx.ap-southeast-
2.amazonaws.com:/sap_hana_dc1_log_shared/lognode1 directory=/hana/log/HDB fstype=nfs
options=defaults,suid op monitor interval=60s on-fail=fence timeout=20s OCF_CHECK_LEVEL=20
clone
[root@dc1hana01]# pcs resource create nfs_hana_log2_dc1 ocf:heartbeat:Filesystem device=svm-
012ab34cd45ef67.fs-0879de29a7fbb752d.fsx.ap-southeast-
2.amazonaws.com:/sap_hana_dc1_log_shared/lognode2 directory=/hana/log/HDB fstype=nfs
options=defaults,suid op monitor interval=60s on-fail=fence timeout=20s OCF_CHECK_LEVEL=20
clone
[root@dc1hana01]# pcs resource create nfs_hana_shared_dc2 ocf:heartbeat:Filesystem
device=svm-012ab34cd45ef78.fs-088e3f66bf4f22c33.fsx.ap-southeast-
2.amazonaws.com:/sap_hana_dc2_log_shared/shared directory=/hana/shared fstype=nfs
options=defaults,suid op monitor interval=60s on-fail=fence timeout=20s OCF_CHECK_LEVEL=20
clone
[root@dc1hana01]# pcs resource create nfs_hana_log_dc2 ocf:heartbeat:Filesystem device=svm-
012ab34cd45ef678.fs-088e3f66bf4f22c33.fsx.ap-southeast-
2.amazonaws.com:/sap_hana_dc2_log_shared/lognode1 directory=/hana/log/HDB fstype=nfs
options=defaults,suid op monitor interval=60s on-fail=fence timeout=20s OCF_CHECK_LEVEL=20
clone
[root@dc1hana01]# pcs resource create nfs_hana_log2_dc2 ocf:heartbeat:Filesystem device=svm-
012ab34cd45ef678.fs-088e3f66bf4f22c33.fsx.ap-southeast-
2.amazonaws.com:/sap_hana_dc2_log_shared/lognode2 directory=/hana/log/HDB fstype=nfs
options=defaults,suid op monitor interval=60s on-fail=fence timeout=20s OCF_CHECK_LEVEL=20
clone
[root@dc1hana01]# pcs node attribute sap-dc1-dbn2 NFS_HDB_SITE=DC1N2
[root@dc1hana01]# pcs node attribute sap-dc2-dbn1 NFS_HDB_SITE=DC2N1
[root@dc1hana01]# pcs node attribute sap-dc2-dbn2 NFS_HDB_SITE=DC2N2
[root@dc1hana01]# pcs node attribute sap-dc1-dbn1 NFS_SHARED_HDB_SITE=DC1
[root@dc1hana01]# pcs node attribute sap-dc1-dbn2 NFS_SHARED_HDB_SITE=DC1
[root@dc1hana01]# pcs node attribute sap-dc2-dbn1 NFS_SHARED_HDB_SITE=DC2
[root@dc1hana01]# pcs node attribute sap-dc2-dbn2 NFS_SHARED_HDB_SITE=DC2
[root@dc1hana01]# pcs constraint location nfs_hana_shared_dc1-clone rule resource-
discovery=never score=-INFINITY NFS_SHARED_HDB_SITE ne DC1
[root@dc1hana01]# pcs constraint location nfs_hana_log_dc1-clone rule resource-discovery=never
score=-INFINITY NFS_HDB_SITE ne DC1N1
[root@dc1hana01]# pcs constraint location nfs_hana_log2_dc1-clone rule resource-discovery=never
score=-INFINITY NFS_HDB_SITE ne DC1N2
[root@dc1hana01]# pcs constraint location nfs_hana_shared_dc2-clone rule resource-
discovery=never score=-INFINITY NFS_SHARED_HDB_SITE ne DC2
```

```
[root@dc1hana01]# pcs constraint location nfs_hana_log_dc2-clone rule resource-discovery=never
score=-INFINITY NFS_HDB_SITE ne DC2N1
[root@dc1hana01]# pcs constraint location nfs_hana_log2_dc2-clone rule resource-discovery=never
score=-INFINITY NFS_HDB_SITE ne DC2N2
[root@dc1hana01]# pcs resource enable nfs_hana_shared_dc1 *[root@dc1hana01]# pcs resource
enable nfs_hana_log_dc1
[root@dc1hana01]# pcs resource enable nfs_hana_log2_dc1
[root@dc1hana01]# pcs resource enable nfs_hana_shared_dc2
[root@dc1hana01]# pcs resource enable nfs_hana_log_dc2
[root@dc1hana01]# pcs resource enable nfs_hana_log2_dc2
[root@dc1hana01]# pcs resource update nfs_hana_shared_dc1-clone meta clone-max=2
interleave=true
[root@dc1hana01]# pcs resource update nfs_hana_shared_dc2-clone meta clone-max=2
interleave=true
[root@dc1hana01]# pcs resource update nfs_hana_log_dc1-clone meta clone-max=1 interleave=true
[root@dc1hana01]# pcs resource update nfs_hana_log_dc2-clone meta clone-max=1 interleave=true
[root@dc1hana01]# pcs resource update nfs_hana_log2_dc1-clone meta clone-max=1
interleave=true
[root@dc1hana01]# pcs resource update nfs_hana_log2_dc2-clone meta clone-max=1
interleave=true
```

4.3. SYSTEMD MANAGED SAP SERVICES

If a **systemd-enabled** SAP HANA version is used (SAP HANA 2.0 SPS07 and later), a shutdown gracefully stops those services. In some environments fencing causes a shutdown and this gracefully stops the service. In some cases, the pacemaker might not work as expected.

If you add drop-in files, then it prevents the service from stopping, for example - **/etc/systemd/system/resource-agents-deps.target.d/sap_systemd_hdb_00.conf**. You can also use other filenames.

```
root@saphdb1:/etc/systemd/system/resource-agents-deps.target.d# more sap_systemd_hdb_00.conf
[Unit]
Description=Pacemaker SAP resource HDB_00 needs the SAP Host Agent service
Wants=saphostagent.service
After=saphostagent.service
Wants=SAPHDB_00.service
After=SAPHDB_00.service
```

These files need to be activated. Use the following command:

```
[root]# systemctl daemon-reload
```

For further information please check Why does the **stop** operation of a SAPHana resource agent fail when the **systemd** based SAP startup framework is enabled? .

4.4. ADDITIONAL HOOKS

Above, you have configured the **srConnectionChanged()** hook. You can also use an additional hook for **srServiceStateChanged()** to manage changes of the **hdbindexserver** processes of SAP HANA instances.

Perform the steps given below to activate the **srServiceStateChanged()** hook for each SAP HANA instance on all HA cluster nodes.

NOTE

This solution is Technology Preview. Red Hat Global Support Services may create bug reports on behalf of subscribed customers who are creating support cases.

Procedure

1. Update the SAP HANA **global.ini** file on each node to enable use of the hook script by both SAP HANA instances (e.g., in file **/hana/shared/RH1/global/hdb/custom/config/global.ini**):

```
[ha_dr_provider_chksrv]
path = /usr/share/SAPHanaSR-ScaleOut
execution_order = 2
action_on_lost = stop

[trace]
ha_dr_saphanasr = info
ha_dr_chksrv = info
```

2. Set the optional parameters as shown below:

   - **action_on_lost** (default: ignore)
   - **stop_timeout** (default: 20)
   - **kill_signal** (default: 9)
     Below is an explanation of the available options for **action_on_lost**:
   - **ignore**: This enables the feature, but only logs events. This is useful for monitoring the hook's activity in the configured environment.
   - **stop**: This executes a graceful **sapcontrol -nr <nr> -function StopSystem**.
   - **kill**: This executes **HDB kill-<signal>** for the fastest stop.

NOTE

stop_timeout is added to the command execution of the stop and kill actions, and **kill_signal** is used in the kill action as part of the **HDB kill-<signal>** command.

3. Reload the **HA/DR** providers to activate the new hook while HANA is running:

```
[rh1adm]$ hdbnsutil -reloadHADRProviders
```

4. Check the new trace file to verify the hook initialization:

```
[rh1adm]$ cdtrace [rh1adm]$ cat nameserver_chksrv.trc
```

For more information, check Implementing a HA/DR Provider .

CHAPTER 5. EXAMPLES AND BEST PRACTICES

5.1. TESTING THE ENVIRONMENT

Perform the following steps to check if everything is working as expected.

Procedure

1. Execute a takeover
   Change the score of the master nodes to do a failover. In this example, the **SAPHana** clone resource is **rsc_SAPHana_HDB_HDB00-clone**, and **saphdb3** is one node in the second availability zone:

   ```
   pcs constraint location rsc_SAPHana_HDB_HDB00-clone rule role=master score=100
   \#uname eq saphdb3
   ```

   This constraint should be removed again with:

   ```
   pcs constraint remove rsc_SAPHana_HDB_HDB00
   ```

   Otherwise, pacemaker tries to start HANA on **SAPHDB1**.

2. Fence a node
   You can fence a node with the command:

   ```
   pcs stonith fence <nodename>
   ```

   Depending on the other fencing options and the infrastructure used, this node will stay down or come back.

3. **kill** HANA
   You can also **kill** the database to check if the SAP resource agent is working. As **sidadm**, you can call:

   ```
   sidadm% HDB kill
   ```

   Pacemaker detects this issue and resolves it with a solution.

5.2. USEFUL ALIASES

5.2.1. Aliases for user root

These aliases are added to **–/.bashrc**:

```
export ListInstances=$(/usr/sap/hostctrl/exe/saphostctrl -function ListInstances| head -1 )
export sid=$(echo "$ListInstances" |cut -d " " -f 5| tr [A-Z] [a-z])
export SID=$(echo $sid | tr [a-z] [A-Z])
export Instance=$(echo "$ListInstances" |cut -d " " -f 7 )
alias crmm='watch -n 1 crm_mon -1Arf'
alias crmv='watch -n 1 /usr/local/bin/crmmv'
alias clean=/usr/local/bin/cleanup
alias cglo='su - ${sid}adm -c cglo'
```

```
alias cdh='cd /usr/lib/ocf/resource.d/heartbeat'
alias vhdbinfo="vim /usr/sap/${SID}/home/hdbinfo;dcp /usr/sap/${SID}/home/hdbinfo"
alias gtr='su - ${sid}adm -c gtr'
alias hdb='su - ${sid}adm -c hdb'
alias hdbi='su - ${sid}adm -c hdbi'
alias hgrep='history | grep $1'
alias hri='su - ${sid}adm -c hri'
alias hris='su - ${sid}adm -c hris'
alias killnode="echo 'b' > /proc/sysrq-trigger"
alias lhc='su - ${sid}adm -c lhc'
alias python='/usr/sap/${SID}/HDB${Instance}/exe/Python/bin/python'
alias pss="watch 'pcs status --full | egrep -e Node\|master\|clone_state\|roles'"
alias srstate='su - ${sid}adm -c srstate'
alias shr='watch -n 5 "SAPHanaSR-monitor --sid=${SID}"'
alias sgsi='su - ${sid}adm -c sgsi'
alias spl='su - ${sid}adm -c spl'
alias srs='su - ${sid}adm -c srs'
alias sapstart='su - ${sid}adm -c sapstart'
alias sapstop='su - ${sid}adm -c sapstop'
alias sapmode='df -h /;su - ${sid}adm -c sapmode'
alias smm='pcs property set maintenance-mode=true'
alias usmm='pcs property set maintenance-mode=false'
alias tma='tmux attach -t 0:'
alias tmkill='tmux killw -a'
alias tm='tail -100f /var/log/messages |grep -v systemd'
alias tms='tail -1000f /var/log/messages | egrep -s\
"Setting master-rsc_SAPHana_${SID}_HDB${Instance}|sr_register\ *|WAITING4LPA\|EXCLUDE as
posible takeover node|SAPHanaSR|failed|${HOSTNAME}\
|PROMOTED|DEMOTED|UNDEFINED|master_walk|SWAIT|WaitforStopped|FAILED"'
alias tmss='tail -1000f /var/log/messages | grep -v systemd\
| egrep -s "secondary with sync status|Settingmaster-rsc_SAPHana_${SID}_HDB${Instance}\
|sr_register|WAITING4LPA|EXCLUDE as posible takeover node|SAPHanaSR\
|failed|${HOSTNAME}|PROMOTED|DEMOTED|UNDEFINED|master_walk|SWAIT|WaitforStopped|FA
ILED"'
alias tmm='tail -1000f /var/log/messages | egrep -s \
"Settingmaster-rsc_SAPHana_${SID}_HDB${Instance}|sr_register\
|WAITING4LPA|PROMOTED|DEMOTED|UNDEFINED|master_walk|SWAIT|W aitforStopped\
|FAILED|LPT|SOK|SFAIL|SAPHanaSR-mon"| grep -v systemd'
alias tmsl='tail -1000f /var/log/messages | egrep -s\
"Settingmaster-rsc_SAPHana_${SID}_HDB${Instance}|sr_register|WAITING4LPA\
|PROMOTED|DEMOTED|UNDEFINED|ERROR|Warning|mast er_walk|SWAIT\
|WaitforStopped|FAILED|LPT|SOK|SFAIL|SAPHanaSR-mon"'
alias vih='vim /usr/lib/ocf/resource.d/heartbeat/SAPHanaStart'
alias switch1='pcs constraint location rsc_SAPHana_HDB_HDB00-clone \
rule role=master score=100 \#uname eq saphdb1'
alias switch3='pcs constraint location rsc_SAPHana_HDB_HDB00-clone \
rule role=master score=100 \#uname eq saphdb3'
alias switch0='pcs constraint remove location-rsc_SAPHana_HDB_HDB00-clone
alias switchl='pcs constraint location | grep pcs resource | grep promotable\
| awk "{ print $4 }"` | grep Constraint| awk "{ print $NF }"'
alias scl='pcs constraint location |grep " Constraint"'
```

5.2.2. Aliases for the SIDadm user

These aliases are added to **~/.customer.sh**:

-

```
alias tm='tail -100f /var/log/messages |grep -v systemd'
alias tms='tail -1000f /var/log/messages | egrep -s \
"Settingmaster-rsc_SAPHana_$SAPSYSTEMNAME_HDB${TINSTANCE}|sr_register\
|WAITING4LPA|EXCLUDE as posible takeover node|SAPHanaSR|failed\
|${HOSTNAME}|PROMOTED|DEMOTED|UNDEFINED|master_walk|SWAIT|WaitforStopped|FAILED"
'
alias tmsl='tail -1000f /var/log/messages | egrep -s \
"Settingmaster-rsc_SAPHana_$SAPSYSTEMNAME_HDB${TINSTANCE}|sr_register\
|WAITING4LPA|PROMOTED|DEMOTED|UNDEFINED|master_walk|SWAIT|WaitforStopped|FAILED|
LPT"'
alias sapstart='sapcontrol -nr ${TINSTANCE} -function StartSystem HDB;hdbi'
alias sapstop='sapcontrol -nr ${TINSTANCE} -function StopSystem HDB;hdbi'
alias sapmode='watch -n 5 "hdbnsutil -sr_state --sapcontrol=1 |grep site.\*Mode"'
alias sapprim='hdbnsutil -sr_stateConfiguration| grep -i primary'
alias sgsi='watch sapcontrol -nr ${TINSTANCE} -function GetSystemInstanceList'
alias spl='watch sapcontrol -nr ${TINSTANCE} -function GetProcessList'
alias splh='watch "sapcontrol -nr ${TINSTANCE} -function GetProcessList\
| grep hdbdaemon"'
alias srs="watch -n 5 'python \
/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/python_support/systemReplicationStatus.py
* *; echo Status \$?'"
alias cdb="cd /usr/sap/${SAPSYSTEMNAME}/HDB${TINSTANCE}/backup"
alias srstate='watch -n 10 hdbnsutil -sr_state'
alias hdb='watch -n 5 "sapcontrol -nr ${TINSTANCE} -function GetProcessList\
| egrep -s hdbdaemon\|hdbnameserver\|hdbindexserver "'
alias hdbi='watch -n 5 "sapcontrol -nr ${TINSTANCE} -function GetProcessList\
| egrep -s hdbdaemon\|hdbnameserver\|hdbindexserver\
;sapcontrol -nr ${TINSTANCE} -function GetSystemInstanceList "'
alias hgrep='history | grep $1'
alias vglo="vim /usr/sap/$SAPSYSTEMNAME/SYS/global/hdb/custom/config/global.ini"
alias vgloh="vim
/hana/shared/${SAPSYSTEMNAME}/HDB${TINSTANCE}/${HOSTNAME}/global.ini"
alias hri='hdbcons -e hdbindexserver "replication info"'
alias hris='hdbcons -e hdbindexserver "replication info" \
| egrep -e "SiteID|ReplicationStatus_"'
alias gtr='watch -n 10 /usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/Python/bin/python \
/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/python_support/getTakeoverRecommendation.
py \
--sapcontrol=1'
alias lhc='/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/Python/bin/python \
/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/python_support/landscapeHostConfiguration.py
\
;echo $?'
alias reg1='hdbnsutil -sr_register --remoteHost=hana07 -remoteInstance=${TINSTANCE} \
--replicationMode=syncmem --name=DC3 --remoteName=DC1 \
--operationMode=logreplay --online'
alias reg2='hdbnsutil -sr_register --remoteHost=hana08 -remoteInstance=${TINSTANCE} \
--replicationMode=syncmem --name=DC3 --remoteName=DC2 \
--operationMode=logreplay --online'
alias reg3='hdbnsutil -sr_register --remoteHost=hana09 -remoteInstance=${TINSTANCE} \
--replicationMode=syncmem --name=DC3 --remoteName=DC3 --operationMode=logreplay \
--online'
PS1="\[\033[m\][\[\e[1;33m\]\u\[\e[1;33m\]\[\033[m\]@\[\e[1;36m\]\h\[\033[m\]: \[\e[0m\]\[\e[1;32m\]\W\
[\e[0m\]]# "
```

5.3. MONITORING FAILOVER EXAMPLE

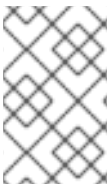
There are many ways to force a takeover. This example forces a takeover without shutting off a node. The SAP resource agents work with scores to decide which node will promote the **SAPHana** clone resource. The current status is seen using this command:

```
[root@saphdb2:~]# alias pss='pcs status --full | egrep -e "Node|master|clone_state|roles"'
[root@saphdb2:~]# pss
Node List:
Node Attributes:
* Node: saphdb1 (1):
* hana_hdb_clone_state : PROMOTED
* hana_hdb_roles : master1:master:worker:master
* master-rsc_SAPHana_HDB_HDB00 : 150
* Node: saphdb2 (2):
* hana_hdb_clone_state : DEMOTED
* hana_hdb_roles : slave:slave:worker:slave
* master-rsc_SAPHana_HDB_HDB00 : -10000
* Node: saphdb3 (3):
* hana_hdb_clone_state : DEMOTED
* hana_hdb_roles : master1:master:worker:master
* master-rsc_SAPHana_HDB_HDB00 : 100
* Node: saphdb4 (4):
* hana_hdb_clone_state : DEMOTED
* hana_hdb_roles : slave:slave:worker:slave
* master-rsc_SAPHana_HDB_HDB00 : -12200
```

In this example, the **SAPHana** clone resource is promoted on **saphdb1**. So the primary database runs on **saphdb1**. The score of this node is **150** and you can adjust the score of the secondary **saphdb3** to force pacemaker to takeover the database to the secondary.

CHAPTER 6. MAINTENANCE PROCEDURES

The following sections describe the recommended procedures to perform maintenance on HA cluster setups used for managing HANA Scale-Out System Replication. You must use these procedures independently from each other.



NOTE

It is not necessary to put the cluster in maintenance-mode when using these procedures. For more information, refer to When to use "maintenance-mode" in RHEL High Availability Add-on for pacemaker based cluster?.

6.1. UPDATING THE OS AND HA CLUSTER COMPONENTS

Please refer to Recommended Practices for Applying Software Updates to a RHEL High Availability or Resilient Storage Cluster, for more information.

6.2. UPDATING THE SAP HANA INSTANCES

Procedure

If the HA cluster configuration described in this document manages the SAP HANA System Replication setup, then you need to execute some additional steps apart from the actual process of updating the SAP HANA instances before and after the update. Execute the following steps:

1. Put the **SAPHana** resource in unmanaged mode:

```
[root]# pcs resource unmanage SAPHana_RH1_HDB10-clone
```

2. Update the SAP HANA instances using the procedure that SAP provides.

3. Refresh the status of the **SAPHana** resource to make sure the cluster is aware of the current state of the SAP HANA System Replication setup when the update of the SAP HANA instances has been completed and it has been verified that SAP HANA System Replication is working again:

```
[root]# pcs resource refresh SAPHana_RH1_HDB10-clone
```

4. Put the **SAPHana** resource back into managed mode so that the HA cluster will be able to react to any issues in the SAP HANA System Replication setup again when the HA cluster correctly picks up the current status of the SAP HANA System Replication setup:

```
[root]# pcs resource manage SAPHana_RH1_HDB10-clone
```

6.3. MOVING SAPHANA RESOURCE TO ANOTHER NODE (SAP HANA SYSTEM REPLICATION TAKEOVER BY HA CLUSTER) MANUALLY

Move the promotable clone resource to trigger a manual takeover of SAP HANA System Replication:

```
[root]# pcs resource move SAPHana_RH1_HDB10-clone
```

NOTE

pcs-0.10.8-1.el8 or later is required for this command to work correctly. For more information, refer to The pcs resource move command fails for a promotable clone unless "--master" is specified.

With each **pcs resource move** command invocation, the HA cluster creates a location constraint to cause the resource to move. For more information, refer to Is there a way to manage constraints when running pcs resource move?.

This constraint must be removed after it has been verified that the SAP HANA System Replication takeover has been completed in order to allow the HA cluster to manage the former primary SAP HANA instance again.

To remove the constraint created by **pcs resource move**, use the following command:

```
[root]# pcs resource clear SAPHana_RH1_HDB10-clone
```

NOTE

What happens to the former SAP HANA primary instance after the takeover has been completed and the constraint has been removed depends on the setting of the **AUTOMATED_REGISTER** parameter of the **SAPHana** resource:

- If **Automated_REGISTER=true**, then the former SAP HANA primary instance is registered as the new secondary, and SAP HANA System Replication becomes active again.
- If **AUTOMATED_REGISTER=false**, then it is up to the operator to decide what should happen with the former SAP HANA primary instance after the takeover.

CHAPTER 7. REFERENCES

7.1. RED HAT

- Configuring RHEL 8 for SAP HANA2 installation
- Configuring and managing high availability clusters
- Support Policies for RHEL High Availability Clusters
- Support Policies for RHEL High Availability Clusters - Fencing/STONITH
- Support Policies for RHEL High Availability Clusters - Management of SAP HANA in a Cluster
- Red Hat HA Solutions for SAP HANA, S/4HANA and NetWeaver based SAP Applications
- Configuring quorum devices
- The Systemd-Based SAP Startup Framework
- Why does the **stop** operation of a SAPHana resource agent fail when the systemd based SAP startup framework is enabled?

7.2. SAP

- SAP HANA Server Installation and Update Guide
- SAP HANA System Replication
- Implementing a HA/DR Provider
- SAP Note 2057595 - FAQ: SAP HANA High Availability
- SAP Note 2063657 - SAP HANA System Replication Takeover Decision Guideline
- SAP Note 2235581 - SAP HANA: Supported Operating Systems
- SAP Note 2369981 - Required configuration steps for authentication with HANA System Replication
- SAP Note 2972496 - SAP HANA Filesystem Types
- SAP Note 3007062 - FAQ: SAP HANA & Third Party Cluster Solutions
- SAP Note 3115048 - sapstartsrv with native Linux systemd support
- SAP Note 3139184 - Linux: systemd integration for sapstartsrv and SAP Host Agent
- SAP Note 3189534 - Linux: systemd integration for sapstartsrv and SAP HANA

7.3. OTHER

- Be Prepared for Using Pacemaker Cluster for SAP HANA – Part 1: Basics
- Be Prepared for Using Pacemaker Cluster for SAP HANA – Part 2: Failure of Both Nodes