



Red Hat Enterprise Linux HA Solution for SAP HANA Scale Out and System Replication

Introduction	3
Target audience	3
Problem description	3
Concepts	4
Overview	4
Supported scenarios	4
Supported responsibilities	4
HANA scale out	5
Master nameserver	6
Scale-out storage configuration	6
Shared file systems: NFS / GPFS	7
Non-shared storage: storage connector	7
HANA system replication	8
Network	9
Pacemaker	11
SAP Hana topology resource agent	13
SAP HANA controller resource agent	13
Preparing the SAP HANA scale-out environment	13
Subscription	13
Network	14
Storage	17
Shared storage	17
Non-shared storage	17
HANA	19
Pacemaker	22
Configuring the SAP HANA scale-out environment	24
Part One: Setting up a basic RHEL configuration	25
Registering your RHEL system and enabling repositories	25
Network configuration settings	27

Configuring /etc/hosts	28
Configuring disks	28
Configuring scale out with shared storage for each datacenter	30
Part Two: Configuring and deploying SAP HANA	31
Configuring SAP base settings	31
Preconfiguring RHEL for SAP HANA	33
Activating the sap-hana tuned profile	35
Disabling automatic numa balancing	37
Disabling Transparent Hugepages (THP)	38
Configuring C-states for lower latency in Linux Intel-based systems	39
Scaling CPU frequency/voltage for Intel-based systems	40
Setting Energy Performance Bias for Intel-based systems	41
Enabling and disabling Kernel Same-page Merging	41
Disabling SELinux	42
Updating systemd to avoid HANA failures	42
Linux: Running SAP applications compiled with GCC 5.6 or 7.	43
Installing the SAP Host Agent	44
Deploying SAP HANA with scale out and system replication	46
Configuring SAP HANA system replication	52
Backing up the primary database	52
Procedure	52
Configuring Pacemaker	58
Installing SAP HANA resource agents for scale-out systems	67
Configuring SAP HANA resource agents	67
Configuring Pacemaker resources	69
Configuring the virtual IP	74
Configuring Pacemaker constraints	74
Summary	81

Introduction

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 scale-out SAP HANA 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.

RHEL for SAP Solutions provides Pacemaker resource agents to automate the takeover from a primary to a secondary data center. This solution provides the concepts, planning, and high-level instructions on how to set up an automated SAP HANA scale-out 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 recommends that setting up and subsequently servicing such a solution should be guided by a certified consultant familiar with both SAP HANA and the Pacemaker-based RHEL High Availability Add-On.

For more information about RHEL for SAP Solutions, see [Red Hat Enterprise Linux for SAP Solutions datasheet](#) and [Overview of the Red Hat Enterprise Linux for SAP Solutions subscription](#).

Target audience

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 7.

Problem description

Building a scale-out SAP HANA environment with system replication over two datacenters includes the manual task of switching the active and passive part of system replication. The decision of when to make this switch is based on multiple data sources and the configuration of the SAP HANA database. For example, in the event of an issue on the primary SAP HANA scale-out instance, you must decide to switch quickly to the replication side or use the scale-out internal cluster functionality. Pacemaker integration analyses the data and configuration and automates the decision based on the environment's configuration.

Concepts

Overview

Building an SAP HANA® scale-out environment with system replication and Pacemaker connectivity combines several complex technologies. SAP HANA configuration is explained in SAP® Notes or documentation that is referenced throughout this document.

An SAP HANA system as a scale-out cluster is primarily used to extend 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 a small High Availability solution.

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 is used by the clients 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.

Supported scenarios

For information about supported scenarios, see [Support Policies for RHEL High Availability Clusters - Management of SAP HANA in a Cluster](#)

Supported responsibilities

Supported by SAP, hardware partners and cloud providers

- Supported hardware and environments
- SAP HANA
- SAP HANA scale-out configuration (SAP cluster setup)
- SAP HANA system replication (SAP cluster setup)

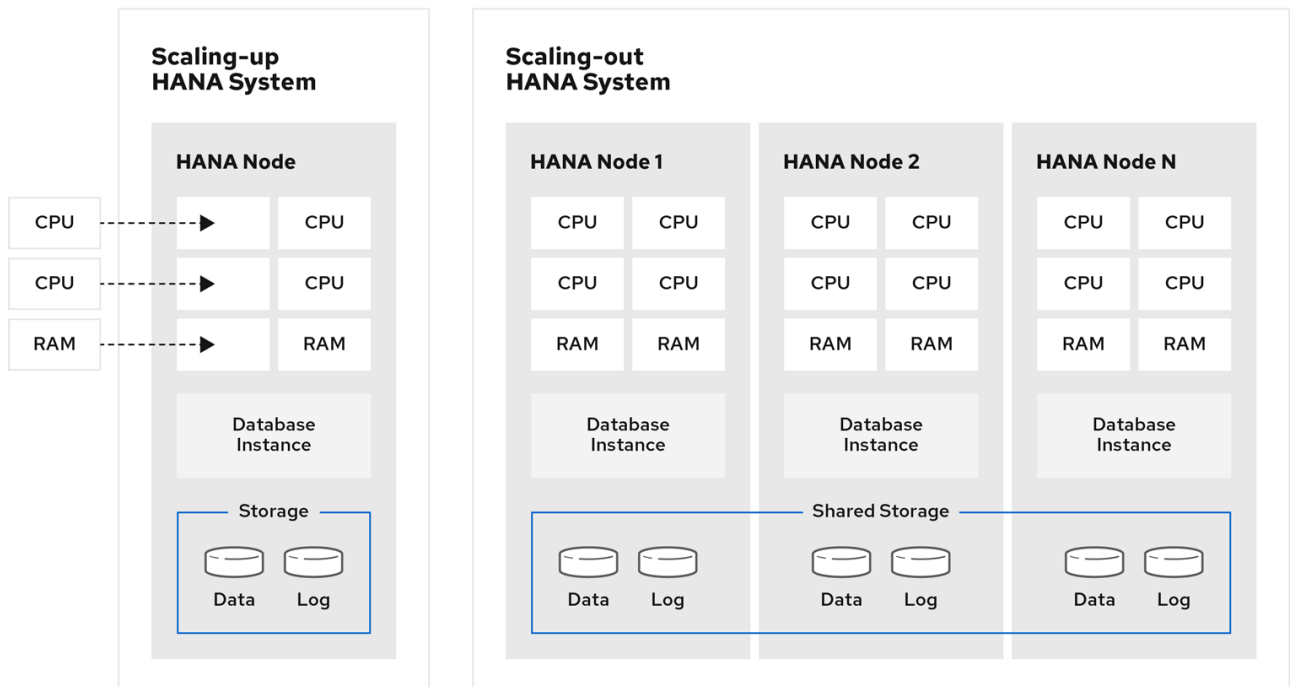
Supported by Red Hat

- 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

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. The prerequisite is a shared storage and a corresponding network connection of 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

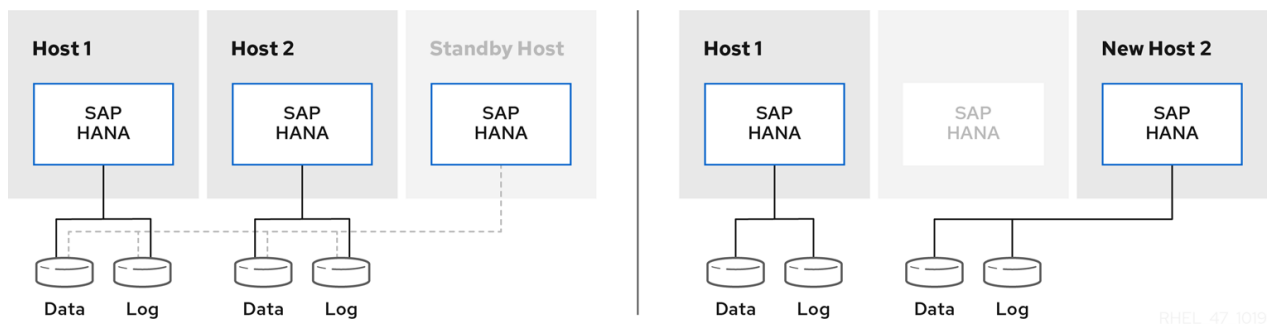


RHEL_47_1019

Master nameserver

A scale-out environment has a master configuration which 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



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 read to take over the process of the failed components.

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

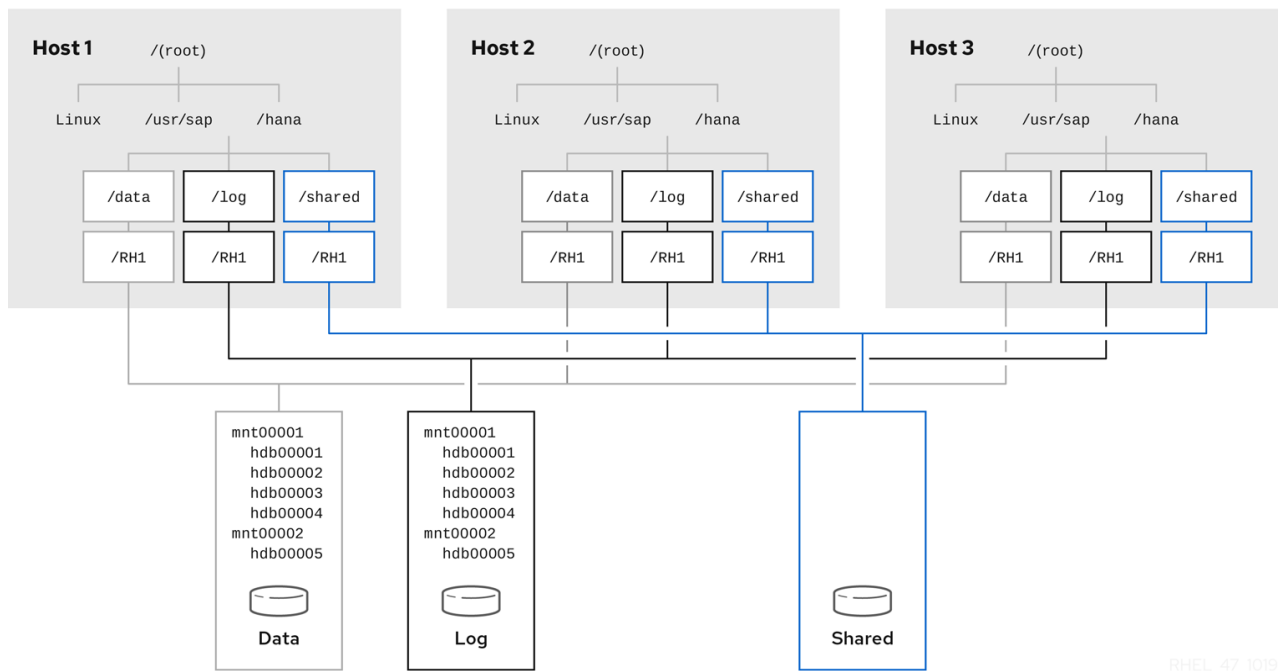
- The first scenario is shared file systems, which offers a file system of all directories over NFS or IBM's GPFS. In this scenario the data is available on all nodes, all the time.
- The second scenario is non-shared storage, which is used to exclusively integrate the required data when needed. All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

Shared file systems: NFS / GPFS

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

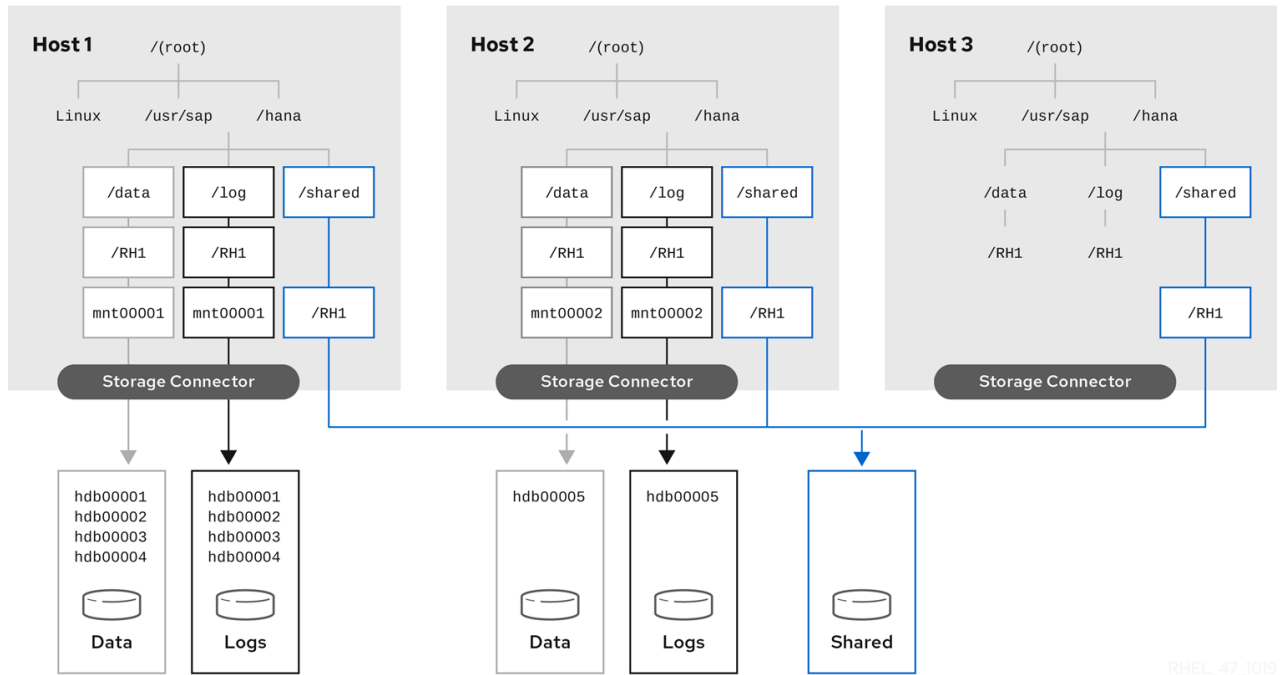


RHEL_47_1019

Non-shared storage: storage connector

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 optimized 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

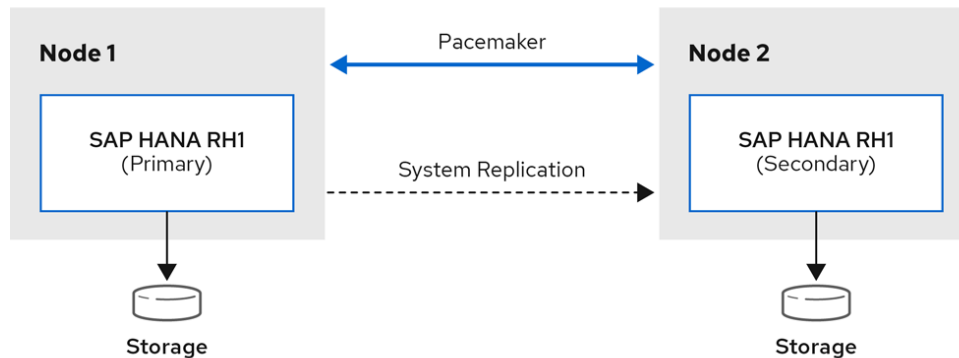


RHEL_47_1019

HANA system replication

With system replication, SAP provides a way for its SAP HANA environment to replicate the database across multiple sites. The data is replicated over the network and is pre-loaded on the second SAP HANA installation. System replication significantly reduces recovery time. When configuring system replication and integrating Pacemaker, it is required that you build up all replicated environments using the same configuration parameters. System replication is compatible with scale-up systems and also with scale-out environments that have different configurations like host-failover configuration.

Figure 5: Overview of Pacemaker integration based on a system replication environment



Network

Three networks are the minimum network requirements for an SAP Hana scale-out and system replication with Pacemaker integration. 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.
- **Pacemaker network:** Required for Pacemaker communication.

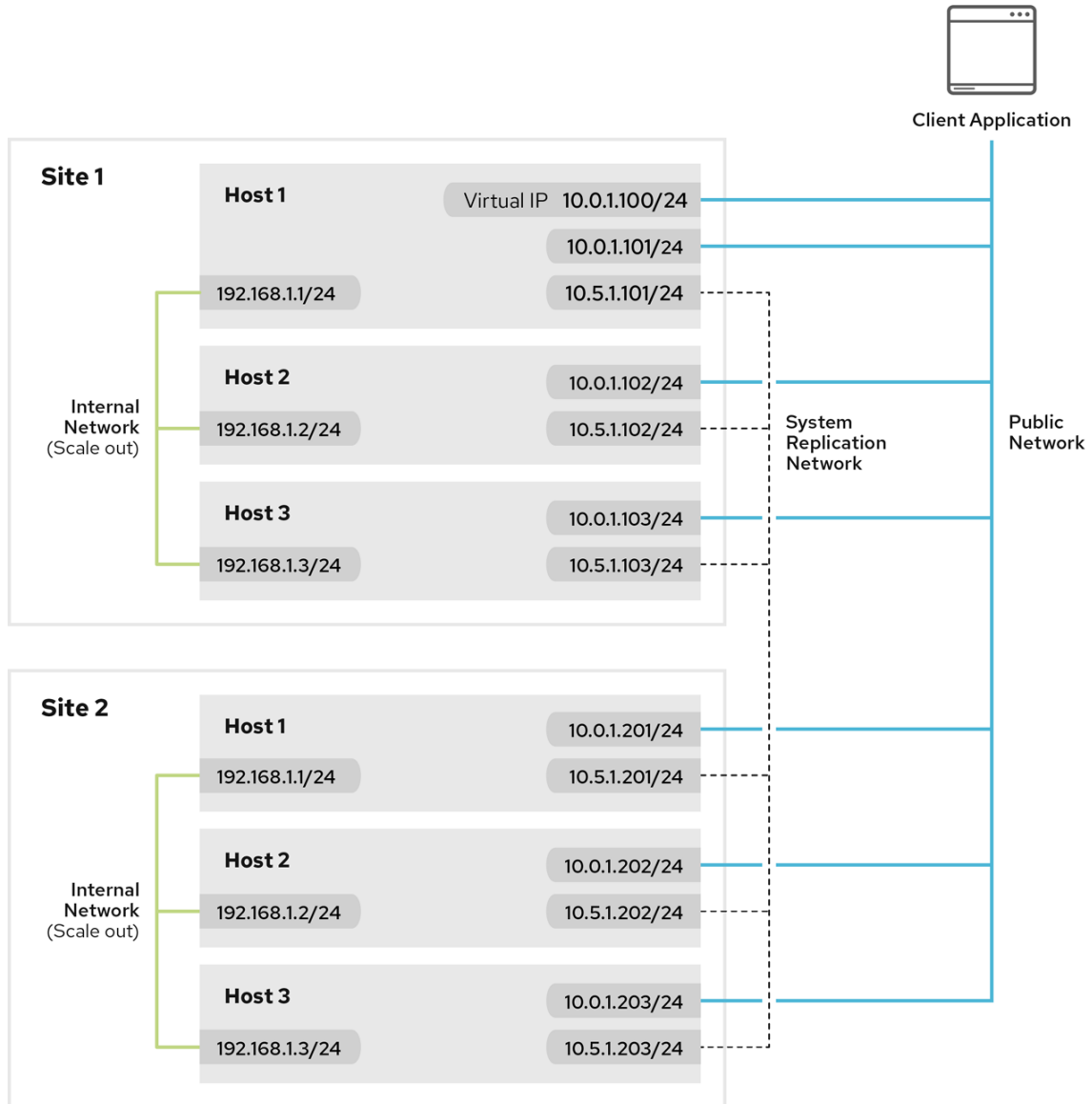
The recommended configuration is designed with the following networks:

- 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 were required. The system replication hostname resolution was adjusted to the network that was used

for the system replication. This is described in the [SAP HANA Network Requirements](#) documentation.

Figure 6: Network configuration of two scale-out systems connected over SAP HANA system replication



Pacemaker

Pacemaker is responsible for ensuring the operation of SAP HANA system replication across two sites. For this reason, resource agents published specifically for SAP HANA scale-out environments are used, which read data from the current system replication and the scale-out environment. After evaluating the data, a decision can be made to either switch the active master node to a scale-out environment or to switch the active side of system replication to the second site. For this solution, a fencing mechanism needed to be configured to avoid split-brain constellations.

For more information about High Availability, see the following documentation:

- [Red Hat Enterprise Linux High Availability Add-On Overview](#)
- [Red Hat Enterprise Linux High Availability Add-On Administration](#)
- [Configuring and managing high availability clusters Red Hat Enterprise Linux 8](#)

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.

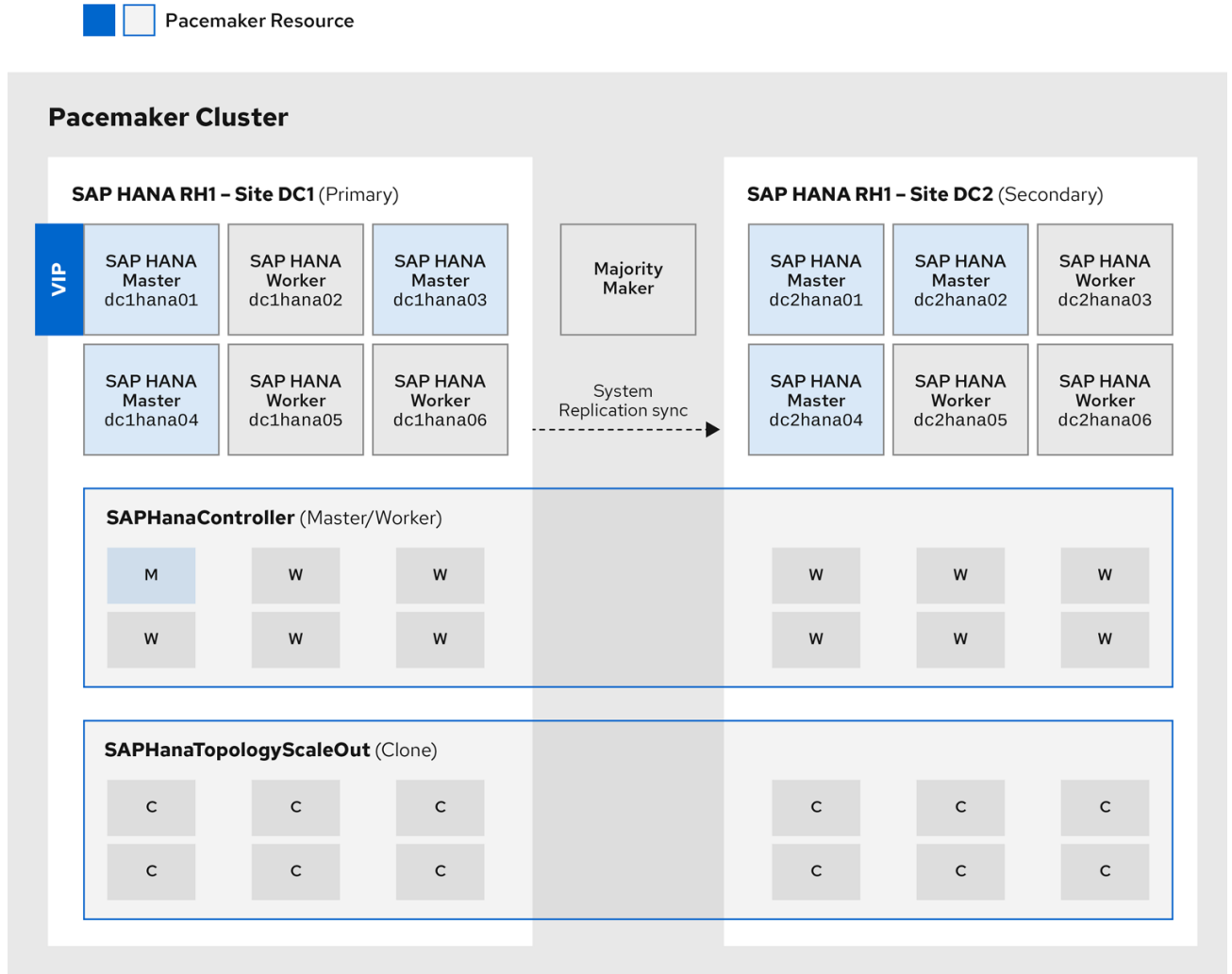
At first, the resource agent is watching for a stable scale-out environment on every site. It will check if enough SAP HANA scale-out master nameserver nodes are configured and in a valid state. Subsequently, the resource agent checks the system replication state. If everything is working correctly, it attaches the virtual IP address to the active master node on the master side of the system replication. In a failure state, the cluster could be 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). STONITH protects data from being corrupted by rogue nodes or concurrent access. An unresponsive node does not mean that it is not accessing data. The only way to be 100% certain that data is safe is to fence the node using STONITH so you can be certain that the node is truly offline, before allowing the data to be accessed from another node. STONITH also has a role to play when a clustered service cannot be stopped. In this solution, the cluster uses STONITH to force the whole node offline, therefore making it safe to start the service elsewhere. To prevent a split-brain scenario, you must integrate an additional cluster

member that works as a majoritymaker. In the event of a network failure, the cluster switches to the primary datacenter.

Figure 7: Example of system replication with scale out



RHEL_47_1019

The cluster configuration is working with two resource agents: one resource agent gathers all the information, and the other resource agent decides what to do with the information.



SAP Hana topology resource agent

The SAP HANA topology resource agent is a cloned resource that receives all of its data from the SAP HANA environment. This data is generated by a configuration process in SAP HANA called "system replication hook." 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 HANA system replication from one side to the other. If the scoring value is higher than a predefined value, the cluster will switch the system replication.

SAP HANA controller resource agent

The SAP HANA controller 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.

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.

Subscription

For an SAP HANA deployment, the following prerequisites are recommended:

- Latest [RHEL for SAP Solutions](#) operating system updates from Red Hat.
- Staging environment with Satellite Server to ensure the correct package versions are installed on every system.

In this solution, connection to the Red Hat network was made without a staging environment. For a pure SAP HANA deployment, the following repository configurations are required:

•

RHEL7	RHEL8
Minimal required repositories	
rhel-7-server-rpms	rhel-8-for-ppc64le-baseos-rpms
rhel-sap-hana-for-rhel-7-server-rpms	rhel-8-for-ppc64le-appstream-rpms
rhel-ha-for-rhel-7-server-rpms	rhel-8-for-ppc64le-highavailability-rpms

Update Services for SAP Solutions (E4S)	
rhel-sap-hana-for-rhel-7-server-e4s-rpms	rhel-8-for-x86_64-baseos-e4s-rpms
rhel-ha-for-rhel-7-server-e4s-rpms	rhel-8-for-x86_64-appstream-e4s-rpms
rhel-sap-hana-for-rhel-7-server-e4s-rpms	rhel-8-for-x86_64-highavailability-e4s-rpms

Red Hat offers Update Services for SAP Solutions on a RHEL for SAP Solutions subscription for customers who want to standardize on a specific minor release for an extended period of time. The Update Services for SAP Solutions allows customers the flexibility to decide when to take advantage of new RHEL features, including new hardware enablement. Each RHEL Update Services for SAP Solutions stream is available for 48 months from the availability of the minor release. For more information, refer to the Update Services for SAP Solutions Add-On section in the [Red Hat Enterprise Linux Life Cycle](#).

Network

SAP recommends building up separate networks for every use case, and Red Hat recommends building up a network for the Pacemaker integration.

SAP recommended networks:

- Application server network
- Client network
- Replication network
- Storage network
- Two internode networks (for scale out)
- Backup network
- Admin network
- Pacemaker

If you are using different scale out and system replication networks, it is important to note that the scale-out network can consist of two different and unconnected scale-out networks. A scale-out SAP HANA cluster configuration is completely self-sufficient, making it possible to configure two separate scale-out networks.

A minimal network environment should include a public network (application server and client network), a network for the SAP HANA communication (replication network, two internode networks), and a Pacemaker network. A Storage network, Backup network, and Admin network



were not required for this solution. In addition to the network configuration, an additional virtual IP is configured using Pacemaker. This IP address is necessary to allow 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: The virtual IP (VIP) address in the public network is needed for communication between the SAP application server and the SAP HANA database, and it is managed by Pacemaker. The following example lists the physical addresses that are mapped to hosts with three nics.

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

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

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.

Shared storage

Shared storage configuration requires method 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`.

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

Datacenter 2

Method	NFS Server	NFS Path	Mount Point
NFS	10.0.1.62/24	/data/dc2/data	/hana/data
NFS	10.0.1.62/24	/data/dc2/log	/hana/log
NFS	10.0.1.62/24	/data/dc2/shared	/hana/shared

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](#).

Non-shared storage example

Parameter	Value
ha_provider	hdb_ha.fcClient

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

Datacenter 2

Method	Parameter Name	WWID
SAN	partition_1_data__wwid	3600508b400105e210001000000491000
SAN	partition_1_log__wwid	3600508b400105e210001000000492000
SAN	partition_2_data__wwid	3600508b400105e210001000000493000
SAN	partition_2_log__wwid	3600508b400105e210001000000494000
Method	Parameter Name	WWID
SAN	partition_3_data__wwid	3600508b400105e210000100000495000
SAN	partition_3_log__wwid	3600508b400105e210000100000496000

Shared devices

Datacenter 1

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

Datacenter 2

Method	NFS Server	NFS Path	Mount Point
NFS	10.0.1.62	/data/dc2/shared	/hana/shared

HANA

There are four steps in building an SAP Hana deployment for a scale-out environment with system replication:

1. Configuring the operating system
2. Installing the SAP Host Agent
3. Deploying two scale-out environments
4. Activating system replication after both SAP HANA databases 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 [softwarecenter](#).

In our example we put the software on a shared directory /install.

Installation sources for SAP HANA Host Agent

Software	Path
Host Agent	/install/51053381/DATA_UNITS/HDB_SERVER_LINUX_X86_64/server/HOSTAGENT.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	Us3YourOwnS3cur3Password
Hostagent SSL certificate password	Us3YourOwnS3cur3Password
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	Us3YourOwnS3cur3Password
HANA components	client,server
System type	Master
System usage	custom
System User Password (SAP)	Us3YourOwnS3cur3Password

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

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

HSR configuration

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

Datacenter 1

Parameter	Value
System replication name	DC1
HSR type	PRIMARY
Parameter	Value
HSR remote host	dc2hana01
SR network	192.168.101.0/24

Datcenter 2

Parameter	Value
System replication name	DC2
HSR type	SECONDARY
HSR remote host	dc1hana01
SR network	192.168.101.0/24

Pacemaker

Pacemaker manages the configuration of SAP HANA, scale out, and 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 [Fencing: Configuring STONITH](#). There are many fence-agents available, please also check:

[yum search fence-agents](#)

Pacemaker configuration is dependent on used hardware or the virtualization environment. In this solution, because Red Hat Virtualization (RHV) is used, the `fence_rhev` fencing method must be configured according to the environment. Also, in this solution, an additional cluster instance as a majoritymaker is required to prevent split-brain scenarios.

Majoritymaker parameters

Hostname	Public Network	Pacemaker
Majoritymaker	10.0.1.42/24	192.168.101.100/24

General parameters

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

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:

1. Setting up a basic RHEL configuration, which is different for every environment.
2. Deploying and configuring SAP HANA scale out for system replication and Pacemaker.

The test environment is built up with eight SAP[®] nodes and an additional `majoritymaker` for Pacemaker configuration. 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. 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 for your SAP HANA scale-out environment](#) section of this document. The following example provides an overview of the required configuration parameters.

Environment

Pacemaker		
4 Nodes (3 + 1)	Majoritymaker	4 Nodes (3 + 1)
Shared Storage (NFS for DC1)	<- System Replication ->	Shared Storage (NFS for DC2)
Network <ul style="list-style-type: none"> - Public Network - HANA Network - Cluster Network 		Network <ul style="list-style-type: none"> - Public Network - HANA Network - Cluster Network

Part One: 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](#) and [2777782](#).

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@lv8081:~]# subscription-manager release
Release: 8.2
[root@lv8081:~]# cat /etc/redhat-release
Red Hat Enterprise Linux release 8.2 (Ootpa)
[root@lv8081:~]#
```

Registering your RHEL system and enabling repositories

Prerequisites

- Install RHEL 7 or RHEL 8
Note: This solution was tested with RHEL 7.6 and RHEL 8.2
- Login as user **root** on every host including the `majoritymaker` for Subscription Management.

Procedure

Note: In this solution, system registration is sent directly to Red Hat as there is no staging configuration. It is recommended that a staging configuration is created 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.

Note: 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 solutions [How to set the hostname in Red Hat Enterprise Linux 7](#) and [How to set hostname using hostnamectl during kickstart in Red Hat Enterprise Linux 7](#). For RHEL 8 please check [Configuring basic system settings Red Hat Enterprise Linux 8](#)

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



```
subscription-manager register
```

2. Enter the username and password.
3. List all pools available with the rhel-sap-hana-for-rhel-7-server-rpms repositories:

```
subscription-manager list --available  
--matches="rhel-sap-hana-for-rhel-7-server-rpms"
```

For RHEL8 please check:

[Configuring basic system settings Red Hat Enterprise Linux 8](#)

For example:

```
subscription-manager list --available  
--matches="rhel-8-for-x86_64-sap-solutions-rpms"
```

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:

```
subscription-manager attach --pool=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

5. Enable the required RHEL7 repositories (RHEL 8 see below):

```
subscription-manager repos --disable="*"
subscription-manager repos --enable=rhel-7-server-rpms
--enable=rhel-sap-hana-for-rhel-7-server-rpms
--enable=rhel-ha-for-rhel-7-server-rpms
```

Note: rhel-7-server-rpms and rhel-sap-hana-for-rhel-7-server-rpms repositories are essential for the deployment and configuration of SAP HANA. The rhel-ha-for-rhel-7-server-rpms repository is part of the Pacemaker integration. Alternatively you can also subscribe to the **Update Services for SAP Solutions (E4S)** repositories.

For RHEL 8 please check if the repos for sap-solutions is enabled:

```
yum repolist | grep sap-solution  
rhel-8-for-x86_64-sap-solutions-rpms      Red Hat Enterprise Linux 8  
for x86_64 - SAP Solutions (RPMs)
```

If not you can enable the for RHEL8 required repos:

```
subscription-manager repos  
--enable=rhel-8-for-x86_64-sap-solutions-rpms  
--enable=rhel-8-for-x86_64-highavailability-rpms
```

For more information, see the [How to register and subscribe a system to the Red Hat Customer Portal using Red Hat Subscription Manager](#) and [How to subscribe SAP HANA systems to the Update Services for SAP Solutions](#) articles.

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

```
yum update -y
```

Network configuration 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 for your SAP HANA scale-out environment](#) section of this document.

```
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
```

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 for your SAP HANA scale-out environment](#) section of this document.

```
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
```

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. For information, see the documentation under *SAP Note 2009879: SAP HANA Guidelines for Red Hat Enterprise Linux (RHEL) Operating System*.

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 file systems are supported using SAP

HANA, for more information, refer to *SAP Note 405827 - Linux: Recommended file systems*. If software RAID is used, the `mkfs.xfs` command automatically configures itself with the correct stripe unit and width to align with the hardware.

2. Create the required mount points:

```
mkdir -p /usr/sap
```

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

```
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 Red article [What are some of best practices for tuning XFS filesystems](#).

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

```
echo "/dev/sdb /usr/sap xfs defaults 1 6" >> /etc/fstab
```

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

```
mount /usr/sap
```

Configuring scale out with shared storage for each datacenter

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 solution, 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 for your 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:

```
yum install -y nfs-utils
```

3. Configure the nodes in Datacenter 1:

```
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
mount -a
```

4. Configure the nodes in Datacenter 2:

```
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
mount -a
```

Part Two: Configuring and deploying SAP HANA

Configuring SAP base settings

Use this procedure to configure SAP base settings. It is necessary to have a preconfigured RHEL system on which are SAP systems are running.

Prerequisites

1. Login as user **root** on every host of the shared storage configuration.
2. Prepare the installation source of SAP HANA.
3. Set the hostname as compatible to SAP HANA:

```
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`:

```
hostname  
<hostname>  
hostname -s  
<hostname>  
hostname -f  
<hostname>.example.com  
hostname -d  
example.com
```

2. Set the system language to **English**:

```
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:

```
yum -y install chrony
systemctl stop chronyd.service
```

Check time server entries;

```
grep ^server /etc/chrony.conf
server 0.de.pool.ntp.org
server 1.de.pool.ntp.org
```

2. Enable and start the `chrony` service:

```
systemctl enable chronyd.service
systemctl start chronyd.service
systemctl restart systemd-timedated.service
```

3. Verify that the `chrony` service is enabled:

```
systemctl status chronyd.service
chronyd.service enabled
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 ^-
1.de.pool.ntp.org 2 8 377 135 -533us[ -533us] +/- 116ms ^-
ntp2.example.com 2 9 377 445 +14ms[ +14ms] +/- 217ms
```




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.

Note: This procedure is based on information in the following SAP Notes:

- *SAP Note 2009879: SAP HANA Guidelines for Red Hat Enterprise Linux (RHEL) Operating System*
- *SAP Note 2292690: SAP HANA DB - Recommended OS settings for RHEL 7*
- *SAP Note 2777782: SAP HANA DB: Recommended OS Settings for RHEL 8.*

The procedure is correct at the time of publishing.

Prerequisite

Login as user **root** on all SAP HANA nodes.

Procedure

1. Install the base package group and other additional required packages required for running SAP HANA on RHEL 7(for RHEL 8 see below):

```
yum -y groupinstall base
yum -y install gtk2 libicu xulrunner sudo tcsh libssh2 expect cairo
graphviz iptraf-ng krb5-workstation krb5-libs libpng12 nfs-utils
lm_sensors rsyslog openssl PackageKit-gtk3-module libcanberra-gtk2
libtool-ltdl xorg-x11-xauth numactl xfsprogs net-tools bind-utils
```

To check the installed group please use:

```
yum grouplist --installed
Installed Environment Groups:
  Server
Installed Groups:
  System Tools
  Container Management
  Headless Management
```

On RHEL8 we recommend to install the Server group:

```
yum -y groupinstall Server
```



On RHEL 8 please use the system roles. More information can be found here: [Red Hat Enterprise Linux System Roles for SAP](#)

2. Add symbolic links (on RHEL 7 only): Some of the library names used during the build process of SAP HANA do not match the library names used on RHEL 7. Therefore, the following symbolic links must be set:

```
ln -s /usr/lib64/libssl.so.10 /usr/lib64/libssl.so.1.0.1
ln -s /usr/lib64/libcrypto.so.10 /usr/lib64/libcrypto.so.1.0.1
```

3. Increase the maximum number of processes a user is allowed to run:

Note: Depending on the workload and the hardware, SAP HANA might start a large amount of processes. In most cases, the maximum number of processes for the `sapsys` group isn't sufficient. The solution is to create the file `/etc/security/limits.d/99-sapsys.conf` with the following content:

```
cat << EOF >> /etc/security/limits.d/99-sapsys.conf
@sapsys soft nproc unlimited
@sapsys hard nproc unlimited
EOF
```

These settings are only activated after the `sapsys` user has been created, for example, during SAP HANA installation. By default, RHEL for SAP HANA is configured to prevent so-called fork bombs. The fact that the group `sapsys` can now create an unlimited number of processes is a potential security issue. After making this change, all users' processes belonging to the `sapsys` group need to be terminated, and users must log out and then log back in for the new setting to take effect.

4. To verify that the new limit is active, log in as a user belonging to the `sapsys` group and run the following command:

```
ulimit -u
unlimited
```

5. All SAP HANA failures are analyzed by SAP Support, and they don't rely on operating system mechanism for failure reporting. To avoid delays when a Linux kernel fails or a core dump occurs, it is possible to disable the application failure and core file handling of the operating system. The `ABRT` service, which handles application failures, can be disabled with the following commands:

```
systemctl disable abrt
systemctl disable abrt-ccpp
```

```
systemctl stop abrttd
systemctl stop abrt-ccpp
```

6. Disable core file creation. To disable core dumps for all users, open `/etc/security/limits.conf`, and add the lines:

```
cat <<EOF >> /etc/security/limits.conf
* soft core 0
* hard core 0
EOF
```

7. By default, RHEL7 enables the kernel failure dump facility (kdump), which can result in longer outages when there is a Linux kernel failure. The system provides a memory dump that can help analyze what caused the kernel to fail. To avoid long outages, it is possible to disable kdump with the following commands:

```
systemctl stop kdump.service
systemctl disable kdump.service
```

8. To protect your SAP HANA servers from unauthorized access, it can be beneficial to configure the built-in firewall of RHEL to only allow access using SAP HANA communication ports. To configure the firewall, use the `firewall-config` tool. In this case, you will need an X11 System to display the GUI version, or you can use the `firewall-config-cmd` command-line tool. To avoid firewall problems during installation, the tool can be disabled completely using the following commands:

```
systemctl stop firewalld
systemctl disable firewalld
```

Activating the sap-hana tuned profile

The sap-hana tuned profile, which is provided by Red Hat as part of RHEL 7 for SAP HANA, contains many of the settings mentioned in the following sections, and it configures some additional settings.

Note: This procedure is based on information in *SAP Note: 2292690: SAP HANA DB: Recommended OS settings for RHEL 7*. The procedure is correct at the time of publishing.

Procedure

1. Activate the `sap-hana` tuned profile on all systems running SAP HANA:

```
yum -y install tuned-profiles-sap-hana
systemctl start tuned
systemctl enable tuned
tuned-adm profile sap-hana
```

2. The `sap-hana` tuned profile must be modified when using version `2.8.0-5.el7_4.2` or earlier on RHEL 7.4, or when using version `2.7.1-3.el7_3.3` or earlier on RHEL 7.3. Copy `/usr/lib/tuned/sap-hana/tuned.conf` to `/etc/tuned/sap-hana/tuned.conf`.

Note: The `/etc/tuned/sap-hana` directory must be created if it doesn't exist.

3. Change the `/etc/tuned/sap-hana/tuned.conf` file as shown:

```
[cpu]
...
force_latency=70
```

Disabling automatic numa balancing

SAP HANA is NUMA (non-uniform memory access)-aware and it does not rely on Linux kernel features to optimize NUMA usage automatically. Therefore, the automatic NUMA balancing features of the Linux kernel should be disabled.

Procedure

1.If you are not using the `sap-hana` tuned profile, create the file `/etc/sysctl.d/sap_hana.conf` and insert the line `kernel.numa_balancing = 0`:

```
echo "kernel.numa_balancing = 0" > /etc/sysctl.d/sap_hana.conf
```

2. Activate the change:

```
sysctl -p /etc/sysctl.d/sap_hana.conf
```

3. Verify that the kernel parameter is set correctly:

```
sysctl kernel.numa_balancing
kernel.numa_balancing = 0
```

4. Disable `numad` if it is installed:

Note: In this solution, `numad` is not installed as it is not part of the base environment group.

```
systemctl status numad
numad.service - numad - The NUMA daemon that manages application
locality.
Loaded: loaded (/usr/lib/systemd/system/numad.service; disabled;
vendor preset: disabled)
Active: inactive (dead)
```

If `numad` is active, disable it:

```
systemctl stop numad
systemctl disable numad
```

Disabling Transparent Hugepages (THP)

SAP HANA can fail occasionally when Transparent Hugepages (THP) is enabled. Use this procedure to disable THP.

Procedure

1. Disable THP during runtime with the following command:

```
echo never > /sys/kernel/mm/transparent_hugepage/enabled
```

2. The `sap-hana` tuned profile deactivates THP. To ensure that THP is deactivated at boot time before the tuned profile is activated, add the following to the line starting with `GRUB_CMDLINE_LINUX` in `/etc/default/grub`:

```
transparent_hugepage=never
```

3. To enable this change, the GRUB2 configuration needs to be rebuilt:

a. On a machine without UEFI, activate this option by issuing:

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

b. On a machine with UEFI, activate this option by issuing:

```
grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

4. Reboot the system. In a scale-out environment, the changes are required on every server in the SAP HANA environment.

5. Verify that THP has been disabled:

```
cat /sys/kernel/mm/transparent_hugepage/enabled  
always madvise [never]
```

Configuring C-states for lower latency in Linux Intel-based systems

The Linux kernel shipped with RHEL 7 includes a `cpuidle` driver for Intel CPUs: `intel_idle`. The `cpuidle` driver causes different behavior in Power-State switching (C-states). The normal operating state is C0 when the processor is put to a higher C state, which saves power. But for low latency applications, the additional time needed to stop and start the execution of the code causes performance degradation.

Procedure

1. The CPU idle states are managed if the `sap-hana` tuned profile is used. If the `sap-hana` tuned profile is not used, it is recommended that you limit the C-states to C0 and C1 by setting the following parameters:

```
processor.max_cstate=1
```

2. Additionally, you should configure the `intel_idle` kernel module to allow C-State C1:

```
intel_idle.max_cstate=1
```

3. You can set this parameter in the kernel command line by modifying `/etc/default/grub`. Append the following parameter to the line starting with `"GRUB_CMDLINE_LINUX"`:

```
processor.max_cstate=1 intel_idle.max_cstate=1
```

4. To enable these changes, the `GRUB2` configuration needs to be rebuilt:

a. On non-UEFI systems run:

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

b. On UEFI systems run:

```
grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

5. Reboot the system. In a scale-out environment, the changes are required on every server of the landscape.

For more information about how to set C-States, see the [What are CPU "C-states" and how to disable them if needed?](#) solution.

Scaling CPU frequency/voltage for Intel-based systems

Linux uses the “CPU governors” power saving technology to control CPU frequency and power consumption. By default, Linux uses the governor “ondemand”, which dynamically scales frequency and voltage up and down depending on CPU load. To maximize CPU performance at the cost of increased energy consumption, SAP recommends the use of the performance governor.

Prerequisite

If you use the `sap-hana` tuned profile, the performance governor is configured using the profile and you don't need to manually change this setting.

Procedure

1.If you are not using the `sap-hana` tuned profile, insert the following code in a system startup script such as `/etc/rc.d/boot.local`:

```
cpupower frequency-set -g performance
```

The setting is applied during system boot. For immediate effect, the `cpupower` command can be run on the command line.

2. Run the following command to check the current settings:

```
cpupower frequency-info
```


Setting Energy Performance Bias for Intel-based systems

The Linux 3.x kernel obeys the Energy Performance Bias (EPB) bit in the CPU flags. Consequently, it configures the CPUs according to the global "energy performance bias" setting in the BIOS.

Procedure

1. It is recommended that the EPB is set using one of the following options:

- Set the EPB in the BIOS to **Maximum performance**.
- Activate the `sap-hana` tuned profile which will set EPB to the **performance governor**.
- Add the following command to a script executed on system boot, such as `/etc/init.d/boot.local`:

```
cpupower set -b 0
```

2. Run `cpupower help set` for more information.

3. Verify that EPB has been set correctly:

```
cpupower info
```

If the command returns the output `perf-bias: 0` then EPB is set to the correct value.

Enabling and disabling Kernel Same-page Merging

The Kernel Same-page Merging (KSM) feature helps reduce physical memory overhead by detecting memory pages with identical content. The feature is useful for virtual machines (VMs), but the space-time tradeoff does not pay off for HDB instances that are not running in VMs. As documented in the [Kernel Same-page Merging \(KSM\) in the 'RHEL 7 Virtualization Tuning and Optimization Guide](#), KSM is normally only activated on RHEL 7 if the system is configured as a virtualization hypervisor.

Procedure

1. If KSM is enabled, it is recommended that you disable it by adding the following command to a script executed on system boot, such as `/etc/init.d/boot.local`:

```
echo "echo 0 > /sys/kernel/mm/ksm/run" >> /etc/init.d/boot.local
```

2. Disable KSM immediately:

```
echo 0 > /sys/kernel/mm/ksm/run
```

3. Verify that KSM is disabled:

```
cat /sys/kernel/mm/ksm/run
```

If the command returns the output '0' then KSM is disabled.

Disabling SELinux

RHEL 7 has SELinux enabled by default. As there is no supported SELinux policy for SAP HANA, leaving SELinux enabled can lead to problems when running SAP HANA on RHEL. SELinux must be disabled to run SAP HANA on RHEL 7.

Procedure

1. For testing purposes SELinux can be set to permissive mode at runtime:

```
setenforce 0  
sed -i  
's/\(SELINUX=enforcing\|SELINUX=permissive\)SELINUX=disabled/g'  
/etc/selinux/config
```

2. Reboot the system and verify that SELinux is fully disabled:

```
sestatus  
SELinux status: disabled  
getenforce  
Disabled
```

Updating systemd to avoid HANA failures

Due to incorrect default settings in the systemd package that is shipped with RHEL7, HANA can fail after installation when the <SID>adm user logs out, or if cron-jobs or other tasks are running for the <SID>adm user end. To avoid this issue, ensure that the systemd package is updated to version systemd-219-19.el7_2.4 or later. Reboot the system after updating the package. For more information, see [Applications using IPC have problems after updating to RHEL 7.2](#) for more information.



Linux: Running SAP applications compiled with GCC 5.6 or 7.

The RPM package `compat-sap-c++` must be installed in addition to the standard compiler runtime libraries. For more information see the documentation:

- *SAP Note 2338763: Linux: Running SAP applications compiled with GCC 5.x*
- *SAP Note 2455582: Linux: Running SAP applications compiled with GCC 6.x*
- *SAP Note 2593824: Linux: Running SAP applications compiled with GCC 7.x*

Prerequisite

To access the `compat-sap-c++` library for SAP HANA, you must have a subscription for RHEL for SAP Solutions. Using the subscription, you can subscribe your server to the "RHEL Server SAP" or "RHEL for SAP HANA" channels on the Red Hat Customer Portal or your local Satellite Server. After subscribing a server to the channel, the output of `yum repolist` must contain the following:

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

Procedure

1. Install the `compat-sap-c++` package based on your SAP HANA version:

- GCC 7 C++ Runtime Compatibility Libraries for SAP HANA 2.0 >= SPS 04:
 - **compat-sap-c++-7 (minimum version: compat-sap-c++-7.el7_2.x86_64)**
 - **libatomic-4.8.5 (min. version: libatomic-4.8.5-36.el7.x86_64)**
- GCC 6 C++ Runtime Compatibility Libraries for SAP HANA 2.0 >= SPS 02:
 - **compat-sap-c++-6 (minimum version: compat-sap-c++-6.el7_2.x86_64)**
- GCC 5 C++ Runtime Compatibility Libraries for SAP HANA 2.0 < SPS 02
 - **Compat-sap-c++-5 (minimum version: compat-sap-c++-5-5.3.1-10)**

For example:

```
yum -y install compat-sap-c++-6
```

2. The minimum version is `compat-sap-c++-6`, for example
`compat-sap-c++-6.e17_2.x86_64`:

```
rpm -q compat-sap-c++-6  
compat-sap-c++-6.e17_2.x86_64
```

3. Reboot the system.

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

- Verify that your `umask` configuration is configured as a standard value; otherwise, the SAP Host Agent installation could fail.
- Login as user **root** on every host for SAP Host Agent installation.

Procedure

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 for your SAP HANA scale-out environment](#) section of this document.

```
adduser sapadm --uid 996  
groupadd sapsys --gid 79  
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:

```
export TMPDIR=$(mktemp -d)  
export  
INSTALLDIRHOSTAGENT=/install/HANA/DATA_UNITS/HDB_SERVER_LINUX_X86_64/
```

```
cp -rp ${INSTALLDIRHOSTAGENT}/server/HOSTAGENT.TGZ $TEMPDIR/  
cd $TEMPDIR  
tar -xzvf HOSTAGENT.TGZ  
cd global/hdb/saphostagent_setup/  
./saphostexec -install
```

Secure operation only works with an encrypted connection. This is achieved by configuring a working SSL connection. An SSL password is required; the following example is based on the parameters in the preparation *for your SAP HANA scale-out environment* section of this document.

```
export MYHOSTNAME=$(hostname)  
export SSLPASSWORD=Us3Your0wnS3cur3Password  
  
export LD_LIBRARY_PATH=/usr/sap/hostctrl/exe/  
export SECUDIR=/usr/sap/hostctrl/exe/sec  

```

3. Verify the SAP Host Agent is available on Port 1129 for all SAP HANA nodes:

```
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](#).

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 [SAP How-to Guide: Network Configuration for SAP HANA System Replication](#).

The SAP HANA database should be installed as described in *SAP Note: 2009879 - SAP HANA Guidelines for Red Hat Enterprise Linux (RHEL) Operating System* and according to the [SAP HANA Server Installation and Update Guide](#).

There are different options to set up the SAP HANA database. The database will be installed on both datacenters with the same SID. A scale out configuration needs at least 3 HANA nodes per side.

The installation per side consists of the following steps:

- Install SAP HANA database on the first node using `hdblcm` (please 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:
`./hdblcm --action=configure_internal_network #` (this is only necessary once)
- Install the other nodes using the shared executable created by the first installation:
For example SID=RH1: `/hana/shared/RH1/hdblcm/hdblcm`
Choose the right host role (worker or standby)
- Repeat the same on the other side
- Setup HANA system replication
 - Copy keys
 - Backup primary database (`SYSTEMDB` and tenant)
 - Stop HANA database on secondary side
 - Register second side as secondary
 - Start HANA database on secondary side

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 is started.

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. In the event of an installation failure, refer to *SAP Note 2009879: SAP HANA Guidelines for Red Hat Enterprise Linux (RHEL) Operating System*.

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 was executed on one node in each datacenter.

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

```
INSTALLDIR=/install/51053381/DATA_UNITS/HDB_SERVER_LINUX_X86_64/  
cd $INSTALLDIR  
./hdblcm --dump_configfile_template=/tmp/templateFile
```

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:

```
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
```

Datcenter 2 example:

```
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. After the installation process is complete, verify that everything is working on one host per datacenter.

```
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

```

[HDBSettings.sh landscapeHostConfiguration.py](#)

```

ok
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 | -       |

```

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/sapstartsrv
pf=/hana/shared/RH1/profile/RH1_HDB10_dc1hana01 -D -u rh1adm

```

Configuring SAP HANA system replication

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

1. Backup the primary database
2. Enable system replication on the primary database
3. Stop secondary database
4. Copy database keys
5. Register the secondary database
6. Start secondary database
7. Verify system replication

This solution provides high-level information about each step. For more in-depth information, see [Setting Up System Replication in the SAP HANA Administration Guide](#).

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, you will 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
mkdir -p /hana/shared/backup/
chown rh1adm /hana/shared/backup/

su - rh1adm
hdbsql -i 10 -u SYSTEM -d SYSTEMDB "BACKUP DATA USING FILE
('/hana/shared/backup/')"
hdbsql -i 10 -u SYSTEM -d RH1 "BACKUP DATA USING FILE
('/hana/shared/backup/')
```

Procedure

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.

```
su - rh1adm
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 on 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. Stop the primary database in the first datacenter (DC1)

```
su - rh1adm
sapcontrol -nr 10 -function StopSystem
```

3. Copy the key and key data file from the primary side 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):

```
scp -rp /usr/sap/RH1/SYS/global/security/rsecssfss/data/SSFS_RH1.DAT
root@dc2hana01:/usr/sap/RH1/SYS/global/security/rsecssfss/data/SSFS_RH
1.DAT
scp -rp /usr/sap/RH1/SYS/global/security/rsecssfss/key/SSFS_RH1.KEY
root@dc2hana01:/usr/sap/RH1/SYS/global/security/rsecssfss/key/SSFS_RH1
.KEY
```

4. After copying both keys to the secondary site, you can register the second datacenter (secondary SAP HANA instance) to the primary SAP HANA instance. 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 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 the SAP HANA system replication a hot standby system. For more information, see the [How To Perform System Replication for SAP HANA Guide](#).

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

```
su - rh1adm
hdbnsutil -sr_register --name=DC2 \
--remoteHost=dc1hana03 --remoteInstance=10 \
--replicationMode=sync --operationMode=logreplay

# Start System
/usr/sap/hostctrl/exe/sapcontrol -nr 10 -function StartSystem
```

After the system starts, run the following commands to verify that everything works as expected. The dispstatus must be green on all nodes (this may take several minutes after initial startup). Also, the landscape host configuration must be in the OK state.

```
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
```

```
rh1adm@dc2hana01:/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 |
| ----- | ----- | ----- | ----- | ----- | |
|---|---|---|---|---|---|
| dc2hana01 | yes   | ok    |        |        | 1 |
1 | default | default | master 1 | master | worker |
master   | worker | worker | default | default |
| dc2hana02 | yes   | ok    |        |        | 2 |
2 | default | default | slave   | slave  | worker |
slave   | worker | worker | default | default |
| dc2hana03 | yes   | ok    |        |        | 3 |
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
rh1adm@dc2hana01:/usr/sap/RH1/HDB10>
```

6. On the Datacenter 1 site, the dispstatus must be green on all nodes and the landscape host configuration must be in the OK state. System replication status is also checked at this point.

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

```
rh1adm@dc1hana01:/hana/shared/backup>
/usr/sap/hostctrl/exe/sapcontrol -nr 10 -function
GetSystemInstanceList
```

```
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:/hana/shared/backup>
```

Aktuellen Landscape Host Configuration anschauen:

HDBSettings.sh landscapeHostConfiguration.py

```
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 |
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

~~~~~

```
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 [SAP Network Configuration for SAP HANA system replication How-to Guide](#).

**Important:** It is recommended that the complete environment is tested to verify all SAP HANA features are working. For more information, see the [SAP HANA Administration Guide for SAP HANA Platform](#).

## Configuring Pacemaker

When the scale-out environment is configured, and system replication is working as expected, Red Hat Pacemaker configuration can be adapted. An additional node is necessary to prevent a Pacemaker split-brain configuration. 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

- Saphostagent is installed: check if `/usr/sap/hostcontrol/exe/sapcontrol` exists  
You can also check [1031096 - Installing Package SAPHOSTAGENT](#)
- Verify that the RHEL High Availability repository is configured in the system. You cannot install Pacemaker without this configuration.
- Login as **root** to all systems.
- To install the packages for the cluster, verify that all cluster nodes are registered and have the required repositories enabled, as described in the *Registering your RHEL system and enabling repositories* section of this document.



- On RHEL 7 Verify that `rhel-x86_64-server-ha-7.X.eus` is attached.

#### `subscription-manager repos --list-enabled`

(Output example is from RHEL 7)

```
+-----+
      Available Repositories in /etc/yum.repos.d/redhat.repo
+-----+
Repository-ID:  rhel-ha-for-rhel-7-server-eus-rpms
Repository-Name: Red Hat Enterprise Linux High Availability (for RHEL
7 Server) - Extended Update Support (RPMs)
Repository-URL:
https://cdn.redhat.com/content/eus/rhel/server/7/7.6/$basearch/highav
ailability/os
Enabled:        1
```

```
Repository-ID:  rhel-7-server-eus-rpms
Repository-Name: Red Hat Enterprise Linux 7 Server - Extended Update
Support (RPMs)
Repository-URL:
https://cdn.redhat.com/content/eus/rhel/server/7/7.6/$basearch/os
Enabled:        1
```

```
Repository-ID:  rhel-sap-hana-for-rhel-7-server-eus-rpms
Repository-Name: RHEL for SAP HANA (for RHEL 7 Server) Extended
Update Support (RPMs)
Repository-URL:
https://cdn.redhat.com/content/eus/rhel/server/7/7.6/$basearch/sap-ha
na/os
Enabled:        1
```

#### `yum repolist`

(Output example is from RHEL 7)

Geladene Plugins: langpacks, product-id, search-disabled-repos,  
subscription-manager

| Repo-ID | Repo-Name: |
|---------|------------|
|---------|------------|

```
Status
rhel-7-server-eus-rpms/x86_64          Red Hat Enterprise
Linux 7 Server - Extended Update Support (RPMs)      24.551
rhel-ha-for-rhel-7-server-eus-rpms/x86_64          Red Hat Enterprise
Linux High Availability (for RHEL 7 Server) - Extended Upd   632
rhel-sap-hana-for-rhel-7-server-eus-rpms/x86_64 RHEL for SAP HANA
(for RHEL 7 Server) Extended Update Support (RPMs)      53
repolist: 25.236
```

## Procedure

1. Configure the cluster. For more information, see [Creating a Red Hat High-Availability Cluster with Pacemaker](#) and [Configuring and managing high availability clusters Red Hat Enterprise Linux 8](#)

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:

```
yum -y install pcs pacemaker fence-agents
```

Alternatively you can also install only specific fence-agents:

```
yum install fence-agents-sbd fence-agents-ipmilan
```

3. If you are running the `firewalld` daemon, execute the following commands to enable the ports that are required by the Red Hat High Availability Add-On:

```
firewall-cmd --permanent --add-service=high-availability
firewall-cmd --add-service=high-availability
```

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

```
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:

```
systemctl start 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.

| RHEL 7.6 and later                                                                                                                    | RHEL 8.0 and later                                                                                                                                                             |
|---------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>pcs cluster auth dc1hana01<br/>dc1hana02 dc1hana03 dc1hana04<br/>dc2hana01 dc2hana02 dc2hana03<br/>dc2hana04 majoritymaker</pre> | <pre>pcs host auth -u hacluster -p<br/>&lt;clusterpassword&gt; dc1hana01<br/>dc1hana02 dc1hana03 dc1hana04<br/>dc2hana01 dc2hana02 dc2hana03<br/>dc2hana04 majoritymaker</pre> |

The output will look similar to:

```
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 command has changed a little bit with RHEL8. The RHEL 8 example also shows, if you are using 2 cluster networks.

| RHEL 7.6 and later                                                                                                                                       | RHEL 8.0 and later                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>pcs cluster setup --start --name hanascaleoutsr dc1hana01 dc1hana02 dc1hana03 dc1hana04 dc2hana01 dc2hana02 dc2hana03 dc2hana04 majoritymaker</pre> | <pre>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</pre> |

The output looks similar to:

```
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:

```
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. Configure STONITH for the environment by utilizing the fence agent that is appropriate for your hardware or virtualization environment. Below is a generic example of configuring a fence device for STONITH:

```
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.

#### `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
```



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:

```
yum install resource-agents-sap-hana-scaleout
```

Verify that the correct Repository is attached. `yum repolist` should contain "rhel-x86\_64-server-sap-hana-<version>" RHEL Server SAP HANA (v. <version> for 64-bit <architecture>).

Configuring SAP HANA resource agents

Use this procedure to preconfigure SAP HANA resource agents.

**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. Configure the hooks by creating a directory in the `hana` shared folder. 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`.

```
mkdir -p /hana/shared/myHooks  
cp /usr/share/SAPHanaSR-ScaleOut/SAPHanaSR.py /hana/shared/myHooks  
chown -R rh1adm:sapsys /hana/shared/myHooks  
  
su - rh1adm  
sapcontrol -nr 10 -function StopSystem  
  
cat <<EOF >> /hana/shared/RH1/global/hdb/custom/config/global.ini  
  
[ha_dr_provider_SAPHanaSR]  
provider = SAPHanaSR  
path = /hana/shared/myHooks  
execution_order = 1
```

```
[trace]
ha_dr_saphanasr = info
EOF
```

3. The cluster configuration requires sudoers configuration on the cluster node. In this solution, this is solved by creating a new file in the `/etc/sudoers.d` directory. Verify that the change is complete for the `<sid>adm` user:

```
cat << EOF > /etc/sudoers.d/20-saphana
# SAPHanaSR-ScaleOut needs for srHook
Cmnd_Alias SOK    = /usr/sbin/crm_attribute -n hana_rh1_glob_srHook -v
SOK    -t crm_config -s SAPHanaSR
Cmnd_Alias SFAIL = /usr/sbin/crm_attribute -n hana_rh1_glob_srHook -v
SFAIL -t crm_config -s SAPHanaSR
rh1adm ALL=(ALL) NOPASSWD: SOK, SFAIL
EOF
```

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

```
# Execute this command on one HANA node in every datacenter
su - rh1adm
sapcontrol -nr 10 -function StartSystem
```

5. Verify that the python script executed by confirming there is a compiled python file in the `myHooks` directory.

```
cd /hana/shared/myHooks
ls -l
-rw-r--r-- 1 rh1adm sapsys 4890 May 4 14:40 SAPHanaSR.py
-rw-r--r-- 1 rh1adm sapsys 4932 Jun 11 15:00 SAPHanaSR.pyc
```

A second verification is to verify the SAP HANA trace log files as `<sid>adm`:

```
cdtrace
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, see the *Monitoring with M\_HA\_DR\_PROVIDERS* section of the [SAP HANA Administration Guide](#).

Configuring Pacemaker resources

The Pacemaker configuration process requires the creation of two resource agents, `SAPHanaTopologyScaleOut` and `SAPHanaController` that control the HANA and Pacemaker environment. Additionally, a virtual IP address is configured 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.

### Prerequisite

- Before starting the configuration process, it is recommended that the cluster is set to maintenance mode to avoid unwanted effects during configuration:

```
pcs property set maintenance-mode=true
```

### Procedure: Configuring the `SAPHanaTopologyScaleOut` resource agent

1. The `SAPHanaTopologyScaleOut` 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 for starting, stopping, and monitoring 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 as **root** on one host in the whole cluster:

| RHEL 7.6 and later                                                                                                                                                                           | RHEL 8.0 and later                                                                                                                                                           |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> pcs resource create rsc_SAPHanaTopology_RH1_HDB10 SAPHanaTopologyScaleOut SID=RH1 InstanceNumber=10 op start timeout=600 op stop timeout=300 op monitor interval=10 timeout=600 </pre> | <pre> pcs resource create rsc_SAPHanaTopology_RH1_HDB10 SAPHanaTopology SID=RH1 InstanceNumber=10 op methods interval=0s timeout=5 op monitor interval=10 timeout=600 </pre> |

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.

```
pcs resource clone rsc_SAPHanaTopology_RH1_HDB10 clone-node-max=1
interleave=true # The same command for RHEL 7 and RHEL 8
```

Once the resource starts, the collected information stored in the form of node attributes can be viewed using the command :

```
pcs status --full
```

### Procedure: Configuring the SAPHanaControl resource agent

When the configuration process for the SAPHanaTopologyScaleOut resource agent is complete, the SAPHanaController resource agent can be configured. While the SAP Hana Topology resource agent collects only data, the SAPHanaTopologyScaleOut 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 InstanceNumber 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 of how to create the `SAPHanaController` master/slave 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:



| RHEL 7.6 and later                                                                                                                                                                                                                                                                                                                                                                       | RHEL 8.0 and later                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> pcs resource create rsc_SAPHana_RH1_HDB10 SAPHanaController SID=RH1 InstanceNumber=10 PREFER_SITE_TAKEOVER=true DUPLICATE_PRIMARY_TIMEOUT=7200 AUTOMATED_REGISTER=true op start interval=0 timeout=3600 op stop interval=0 timeout=3600 op promote interval=0 timeout=3600 op monitor interval=60 role="Master" timeout=700 op monitor interval=61 role="Slave" timeout=700 </pre> | <pre> 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="Master" timeout=700 op monitor interval=61 role="Slave" timeout=700 op promote interval=0 timeout=3600 op start interval=0 timeout=3600 op stop interval=0 timeout=3600 </pre> |
| <pre> pcs resource master msl_rsc_SAPHana_RH1_HDB10 rsc_SAPHana_RH1_HDB10 master-max="1" clone-node-max=1 interleave=true </pre>                                                                                                                                                                                                                                                         | <pre> pcs resource promotable rsc_SAPHana_RH1_HDB10 promoted-max=1 clone-node-max=1 interleave=true </pre>                                                                                                                                                                                                                                                                                                                                                  |

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

For more information, see [Multi-State Resources: Resources That Have Multiple Modes](#)

### Configuring the virtual IP

The SAP HANA environment cluster contains a virtual IP address to reach the master instance of SAP HANA. The following command is an example of how to create a `IPaddr2` resource with the virtual IP 10.0.0.250:

```
pcs resource create rsc_ip_SAPHana_RH1_HDB10 ocf:heartbeat:IPaddr2
ip=10.0.0.250 op monitor interval="10s" timeout="20s"
```

### Configuring Pacemaker constraints

For correct operation, verify that `SAPHanaTopologyScaleOut` resources are started before starting the `SAPHanaController` resources, and also that the virtual IP address is present on the node where the master resource of `SAPHanaController` is running. 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.

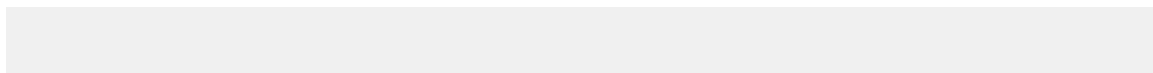
1. Create the constraint:

| RHEL 7.6 and later                                                                                 | RHEL 8.0 and later                                                                                               |
|----------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| <pre>pcs constraint order rsc_SAPHanaTopology_RH1_HDB10-clone then msl_rsc_SAPHana_RH1_HDB10</pre> | <pre>pcs constraint order start rsc_SAPHanaTopology_RH1_HDB10-clone then start rsc_SAPHana_RH1_HDB10-clone</pre> |

2. Colocate the `IPaddr2` resource with the master `SAPHana` resource:

| RHEL 7.6 and later                                                                                        | RHEL 8.0 and later                                                                                          |
|-----------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| <code>pcs constraint colocation add rsc_ip_SAPHana_RH1_HDB10 with master msl_rsc_SAPHana_RH1_HDB10</code> | <code>pcs constraint colocation add rsc_ip_SAPHana_RH1_HDB10 with master rsc_SAPHana_RH1_HDB10-clone</code> |

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



| RHEL 7.6 and later                                                                            | RHEL 8.0 and later                                                                            |
|-----------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| <code>pcs constraint location msl_rsc_SAPHana_RH1_HDB10 avoids majoritymaker</code>           | <code>pcs constraint location rsc_SAPHana_RH1_HDB10-clone avoids majoritymaker</code>         |
| <code>pcs constraint location rsc_SAPHanaTopology_RH1_HDB10-clone avoids majoritymaker</code> | <code>pcs constraint location rsc_SAPHanaTopology_RH1_HDB10-clone avoids majoritymaker</code> |

4. Disable maintenance mode:



5. Verify the cluster environment is working correctly by running the following three commands:

- `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.

**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 ]
Master/Slave Set: ms1_rsc_SAPHana_RH1_HDB10 [rsc_SAPHana_RH1_HDB10]
  Masters: [ dc1hana01 ]
  Slaves: [ 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
```

**pcs status --full**

```
Cluster name: hanascaleoutsr
```

```
Stack: corosync
Current DC: majoritymaker (9) (version 1.1.18-11.el7_5.4-2b07d5c5a9)
- partition with quorum
Last updated: Tue Mar 26 16:34:22 2019
Last change: Tue Mar 26 16:34:03 2019 by root via crm_attribute on
dc2hana01
```

```
9 nodes configured
20 resources configured
```

```
Online: [ dc1hana01 (1) dc1hana02 (2) dc1hana03 (3) dc1hana04 (4)
dc2hana01 (5) dc2hana02 (6) dc2hana03 (7) dc2hana04 (8) majoritymaker
(9) ]
```

```
Full list of resources:
```

```
Clone Set: rsc_SAPHanaTopology_RH1_HDB10-clone
[rsc_SAPHanaTopology_RH1_HDB10]
  rsc_SAPHanaTopology_RH1_HDB10
(ocf::heartbeat:SAPHanaTopology): Started dc2hana02
  rsc_SAPHanaTopology_RH1_HDB10
(ocf::heartbeat:SAPHanaTopology): Started dc1hana03
  rsc_SAPHanaTopology_RH1_HDB10
(ocf::heartbeat:SAPHanaTopology): Started dc2hana04
  rsc_SAPHanaTopology_RH1_HDB10
(ocf::heartbeat:SAPHanaTopology): Started dc2hana03
  rsc_SAPHanaTopology_RH1_HDB10
(ocf::heartbeat:SAPHanaTopology): Started dc1hana04
  rsc_SAPHanaTopology_RH1_HDB10
(ocf::heartbeat:SAPHanaTopology): Started dc1hana01
  rsc_SAPHanaTopology_RH1_HDB10
(ocf::heartbeat:SAPHanaTopology): Started dc1hana02
  rsc_SAPHanaTopology_RH1_HDB10
(ocf::heartbeat:SAPHanaTopology): Started dc2hana01
  rsc_SAPHanaTopology_RH1_HDB10
(ocf::heartbeat:SAPHanaTopology): Stopped
  Started: [ dc1hana01 dc1hana02 dc1hana03 dc1hana04 dc2hana01
dc2hana02 dc2hana03 dc2hana04 ]
```

```

    Stopped: [ majoritymaker ]
Master/Slave Set: ms1_rsc_SAPHana_RH1_HDB10 [rsc_SAPHana_RH1_HDB10]
    rsc_SAPHana_RH1_HDB10      (ocf::heartbeat:SAPHanaController):
Slave dc2hana02
    rsc_SAPHana_RH1_HDB10      (ocf::heartbeat:SAPHanaController):
Slave dc1hana03
    rsc_SAPHana_RH1_HDB10      (ocf::heartbeat:SAPHanaController):
Slave dc2hana04
    rsc_SAPHana_RH1_HDB10      (ocf::heartbeat:SAPHanaController):
Slave dc2hana03
    rsc_SAPHana_RH1_HDB10      (ocf::heartbeat:SAPHanaController):
Slave dc1hana04
    rsc_SAPHana_RH1_HDB10      (ocf::heartbeat:SAPHanaController):
Slave dc1hana01
    rsc_SAPHana_RH1_HDB10      (ocf::heartbeat:SAPHanaController):
Slave dc1hana02
    rsc_SAPHana_RH1_HDB10      (ocf::heartbeat:SAPHanaController):
Master dc2hana01
    rsc_SAPHana_RH1_HDB10      (ocf::heartbeat:SAPHanaController):
Stopped
    Masters: [ dc2hana01 ]
    Slaves: [ dc1hana01 dc1hana02 dc1hana03 dc1hana04 dc2hana02
dc2hana03 dc2hana04 ]
    Stopped: [ majoritymaker ]
    rsc_ip_SAPHana_RH1_HDB10    (ocf::heartbeat:IPAddr2):
Started dc2hana01
    fencing      (stonith:fence_rhevm): Started majoritymaker

Node Attributes:
* Node dc1hana01 (1):
    + hana_rh1_clone_state      : DEMOTED
    + hana_rh1_roles            :
master2:master:worker:master
    + hana_rh1_site              : DC1
    + master-rsc_SAPHana_RH1_HDB10 : 100
* Node dc1hana02 (2):
    + hana_rh1_clone_state      : DEMOTED
    + hana_rh1_roles            : slave:slave:worker:slave

```

```

+ hana_rh1_site : DC1
+ master-rsc_SAPHana_RH1_HDB10 : -12200
* Node dc1hana03 (3):
+ hana_rh1_clone_state : DEMOTED
+ hana_rh1_roles :
master3:slave:worker:standby
+ hana_rh1_site : DC1
+ master-rsc_SAPHana_RH1_HDB10 : 80
* Node dc1hana04 (4):
+ hana_rh1_clone_state : DEMOTED
+ hana_rh1_roles : master1:slave:standby:slave
+ hana_rh1_site : DC1
+ master-rsc_SAPHana_RH1_HDB10 : 80
* Node dc2hana01 (5):
+ hana_rh1_clone_state : PROMOTED
+ hana_rh1_roles :
master1:master:worker:master
+ hana_rh1_site : DC2
+ master-rsc_SAPHana_RH1_HDB10 : 150
* Node dc2hana02 (6):
+ hana_rh1_clone_state : DEMOTED
+ hana_rh1_roles : master3:slave:worker:slave
+ hana_rh1_site : DC2
+ master-rsc_SAPHana_RH1_HDB10 : 110
* Node dc2hana03 (7):
+ hana_rh1_clone_state : DEMOTED
+ hana_rh1_roles : slave:slave:worker:slave
+ hana_rh1_site : DC2
+ master-rsc_SAPHana_RH1_HDB10 : -10000
* Node dc2hana04 (8):
+ hana_rh1_clone_state : DEMOTED
+ hana_rh1_roles :
master1:slave:standby:standby
+ hana_rh1_site : DC2
+ master-rsc_SAPHana_RH1_HDB10 : 115
* Node majoritymaker (9):
+ hana_rh1_roles : :shtdown:shtdown:shtdown

```

Migration Summary:

- \* Node dc2hana02 (6):
- \* Node majoritymaker (9):
- \* Node dc1hana03 (3):
- \* Node dc2hana04 (8):
- \* Node dc2hana03 (7):
- \* Node dc1hana04 (4):
- \* Node dc1hana01 (1):
- \* Node dc1hana02 (2):
- \* Node dc2hana01 (5):

PCSD Status:

dc1hana01: Online  
dc2hana03: Online  
dc2hana04: Online  
dc1hana03: Online  
dc2hana01: Online  
majoritymaker: Online  
dc2hana02: Online  
dc1hana04: Online  
dc1hana02: Online

Daemon Status:

corosync: active/enabled  
pacemaker: active/enabled  
pcsd: active/enabled

**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 master1:master:worker:master 150 DC1
2 DEMOTED master2:slave:worker:slave 110 DC1
3 DEMOTED slave:slave:worker:slave -10000 DC1
4 DEMOTED master3:slave:standby:standby 115 DC1
5 DEMOTED master2:master:worker:master 100 DC2
6 DEMOTED master3:slave:worker:slave 80 DC2
7 DEMOTED slave:slave:worker:slave -12200 DC2
8 DEMOTED master1:slave:standby:standby 80 DC2
9 :shutdown:shutdown:shutdown
```

## Summary

RHEL for SAP Solutions can be used to enhance native SAP HANA replication and failover concepts by providing resource agents to automate the takeover from a primary to a secondary data center.

This document provides information on planning and implementing automated takeover for SAP HANA scale-out deployments. For example:

- Steps to set up an SAP HANA scale-out environment with SAP HANA system replication configured.
- SAP HANA system replication in this solution provides continuous synchronization between two SAP HANA databases to support high availability and disaster recovery.

While this solution has been extensively tested and is known to work, the challenges of a real implementation are typically more complex than can be covered in upfront testing. Please ensure that you adapt all examples provided in this document to your environment before use.

If you are not familiar with the commands used throughout this document, please involve an expert. 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.