



# Red Hat Enterprise Linux for SAP Solutions 8

**Configuring a Cost-Optimized SAP S/4HANA  
HA cluster (HANA System Replication +  
ENSA2) using the RHEL HA Add-On**



# Red Hat Enterprise Linux for SAP Solutions 8 Configuring a Cost-Optimized SAP S/4HANA HA cluster (HANA System Replication + ENSA2) using the RHEL HA Add-On

---

## Legal Notice

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

Describes the setup of a two-node HA cluster for managing both HANA System Replication and Standalone Enqueue Server 2 (ENSA2) for SAP S/4HANA 1809 and later.

## Table of Contents

<b>MAKING OPEN SOURCE MORE INCLUSIVE</b> .....	<b>4</b>
<b>PROVIDING FEEDBACK ON RED HAT DOCUMENTATION</b> .....	<b>5</b>
<b>CHAPTER 1. OVERVIEW</b> .....	<b>6</b>
1.1. INTRODUCTION	6
1.2. AUDIENCE	6
1.3. CONCEPTS	7
1.4. SUPPORT POLICIES	7
<b>CHAPTER 2. REQUIREMENTS</b> .....	<b>8</b>
2.1. SUBSCRIPTION	8
2.2. PACEMAKER RESOURCE AGENTS	8
2.2.1. SAPInstance	8
2.2.2. SAPHanaTopology (Cloned Resource)	8
2.2.3. SAPHana (Promotable Cloned resource)	8
2.2.4. filesystem	8
2.2.5. IPAddr2 (or other RAs for managing VIPs on CCSPs)	8
2.3. TWO NODE CLUSTER ENVIRONMENT	8
2.4. STORAGE REQUIREMENTS	9
2.4.1. Instance Specific Directory	9
2.4.2. Shared Directories	9
<b>CHAPTER 3. INSTALL SAP S/4HANA</b> .....	<b>11</b>
3.1. CONFIGURATION OPTIONS USED IN THIS DOCUMENT	11
3.2. PREPARE HOSTS	11
3.3. INSTALL S/4HANA	11
3.3.1. Install S/4 on s4node1	12
3.3.2. Install ERS on s4node2	12
3.3.3. SAP HANA	13
3.4. POST INSTALLATION	14
3.4.1. ASCS profile modification	14
3.4.2. ERS profile modification	14
3.4.3. Update the /usr/sap/sapservices file	14
3.4.4. Create mount points for ASCS and ERS on the failover node	14
3.4.5. Manually Testing Instances on Other Node	14
3.4.6. Check SAP HostAgent on all nodes	15
3.4.7. Install permanent SAP license keys	15
<b>CHAPTER 4. INSTALL PACEMAKER</b> .....	<b>16</b>
4.1. CONFIGURE GENERAL CLUSTER PROPERTIES	16
4.2. INSTALL RESOURCE-AGENTS-SAP ON ALL CLUSTER NODES	16
4.3. CONFIGURE CLUSTER RESOURCES FOR SHARED FILESYSTEMS	16
4.3.1. Configure shared filesystems managed by the cluster	17
4.3.2. Configure shared filesystems managed outside of cluster	17
4.4. CONFIGURE ASCS RESOURCE GROUP	17
4.4.1. Create resource for virtual IP address	17
4.4.2. Create resource for ASCS filesystem.	17
4.4.3. Create resource for ASCS instance	18
4.5. CONFIGURE ERS RESOURCE GROUP	18
4.5.1. Create resource for virtual IP address	18
4.5.2. Create resource for ERS filesystem	18

4.5.3. Create resource for ERS instance	19
4.6. CREATE CONSTRAINTS	19
4.6.1. Create colocation constraint for ASCS and ERS resource groups	19
4.6.2. Create location constraint for ASCS resource	19
4.6.3. Create order constraint for ASCS and ERS resource groups	19
4.6.4. Create order constraint for /sapmnt resource managed by cluster	20
<b>CHAPTER 5. TEST THE CLUSTER CONFIGURATION</b>	<b>21</b>
5.1. CHECK THE CONSTRAINTS	21
5.2. FAILOVER ASCS DUE TO NODE CRASH	21
5.3. ERS MOVES TO THE PREVIOUSLY FAILED NODE	22
<b>CHAPTER 6. ENABLE CLUSTER TO AUTO-START AFTER REBOOT</b>	<b>23</b>
<b>CHAPTER 7. TEST FAILOVER</b>	<b>24</b>
7.1. FAILOVER ASCS DUE TO NODE CRASH	24
7.2. ERS REMAINS ON CURRENT NODE	24
7.3. TEST ERS CRASH	24
7.4. YOUR COST-OPTIMIZED SAP S/4HANA CLUSTER ENVIRONMENT CAN LOOK LIKE THE ONE BELOW	24
<b>CHAPTER 8. OPTIONAL - ENABLE SAP HA INTERFACE FOR MANAGEMENT OF CLUSTER-CONTROLLED ASCS/ERS INSTANCES USING SAP MANAGEMENT TOOLS</b>	<b>26</b>



## MAKING OPEN SOURCE MORE INCLUSIVE

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



## PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

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

### Submitting feedback through Jira (account required)

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

# CHAPTER 1. OVERVIEW

## 1.1. INTRODUCTION

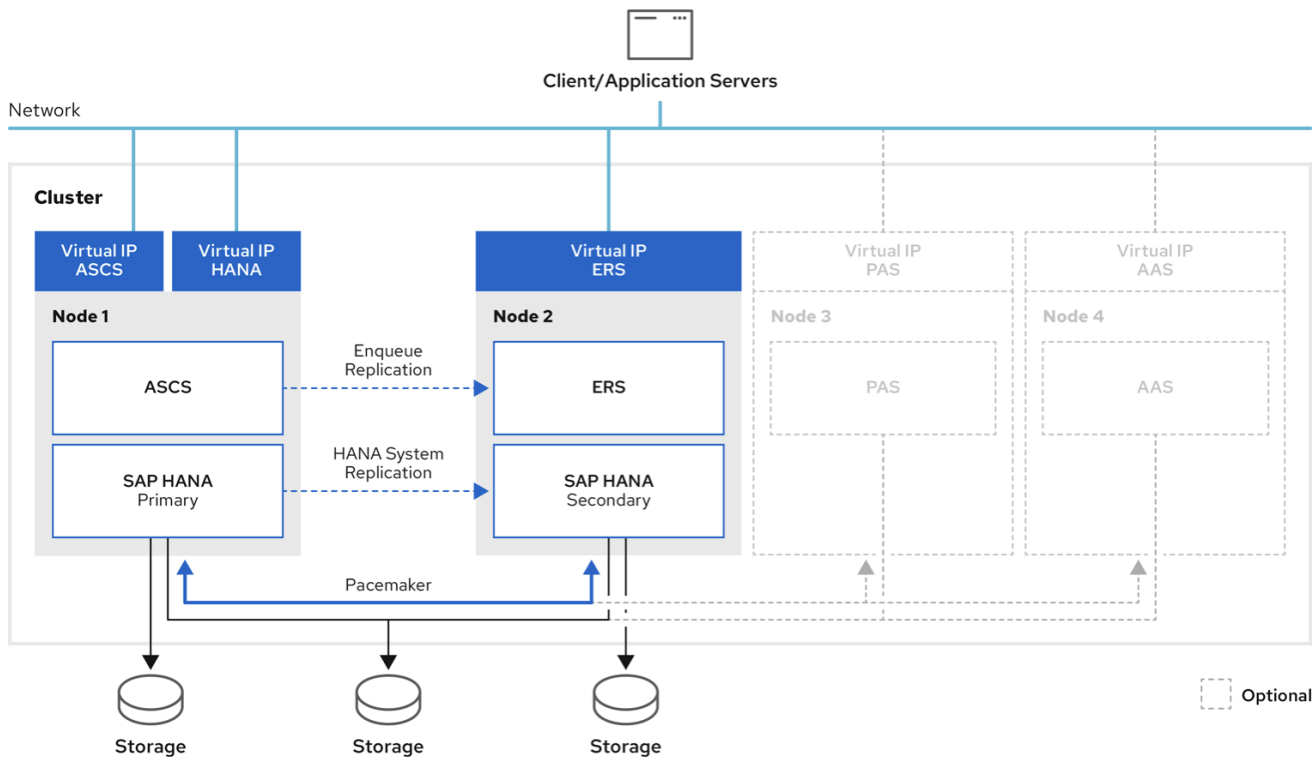
Cost-Optimized deployments of SAP S/4HANA systems play an important role in S/4HANA migration scenarios, especially in saving costs related to additional nodes. It is also critical if such systems are to be made highly available, in which case, the constraints need to be correctly configured.

A typical Cost-Optimized setup for SAP S/4HANA High-Availability consists of 2 distinct components:

- SAP S/4HANA ASCS/ERS cluster resources
- Database: SAP HANA with System Replication
  - Please refer to [Configure Automated HANA System Replication in pacemaker](#) , for more details.

This article focuses on the setup of SAP S/4HANA HA environment where both SAP HANA System Replication and the ASCS and ERS instances are managed by a single cluster. This is done using the RHEL HA add-on and the corresponding HA solutions for SAP, available as part of RHEL for SAP Solutions.

**Note:** Below is the architecture diagram of the example installation of a 2-node cluster setup which this article focuses on, with a separate section on the design and configuration of the additional SAP HANA instances with System Replication. Note that ASCS or SAP HANA primary instances can failover to the other node independently of each other.



## 1.2. AUDIENCE

This document is intended for SAP and Red Hat certified or trained administrators and consultants who already have experience setting up highly available solutions using the Red Hat Enterprise Linux (RHEL) HA add-on or other clustering solutions. Access to both SAP Service Marketplace and Red Hat Customer Portal is required to be able to download software and additional documentation.

Red Hat Professional Services is highly recommended to set up the cluster and customize the solution to meet the customer's data center requirements, which may be different than the solution presented in this document.

## 1.3. CONCEPTS

This document describes how to set up a Cost-Optimized, two-node cluster solution that conforms to the high availability guidelines established by SAP and Red Hat. It is based on Standalone Enqueue Server 2 (ENSA2), now the default installation in SAP S/4HANA 1809 or newer, on top of RHEL 8 for SAP Solutions or above, and highlights a scale-up SAP HANA instance that supports fully automated failover using SAP HANA System Replication. According to SAP, ENSA2 is the successor to Standalone Enqueue Server 1 (ENSA1). It is a component of the SAP lock concept and manages the lock table. This principle ensures the consistency of data in an ABAP system. During a failover with ENSA1, the ASCS instance is required to "follow" the Enqueue Replication Server (ERS). That is, the HA software had to start the ASCS instance on the host where the ERS instance is currently running. In contrast to ENSA1, the newer ENSA2 model and Enqueue Replicator 2 no longer have these restrictions. For more information on ENSA2, please refer to SAP OSS Note [2630416 - Support for Standalone Enqueue Server 2](#).

Additionally, the document will also highlight the SAP HANA Scale-Up instance, with fully automated failover using SAP HANA System Replication, where the SAP HANA promotable clone resources will run on each node as per set constraints. This article does **NOT** cover preparation of the RHEL system for SAP HANA installation, nor the SAP HANA installation procedure. For fast and error-free preparation of the systems for SAP S/4HANA and SAP HANA, we recommend using [RHEL System Roles for SAP](#).

Configuration of both is considered as a Cost-Optimized SAP S/4HANA with an Automated SAP HANA Scale-Up System Replication Environment.

## 1.4. SUPPORT POLICIES

Please refer to [Support Policies for RHEL High Availability Clusters - Management of SAP S/4HANA](#) and [Support Policies for RHEL High Availability Clusters - Management of SAP HANA in a Cluster](#) for more details.

This solution is supported subject to fulfilling the above policies.

## CHAPTER 2. REQUIREMENTS

### 2.1. SUBSCRIPTION

It's important to keep the subscription, kernel, and patch level identical on all cluster nodes.

To be able to use this HA solution, either the [RHEL for SAP Solutions](#) (for on-premise or BYOS setups in public cloud environments) or [RHEL for SAP with High Availability and Update Services](#) (when using PAYG in public cloud environments) subscriptions are required for all cluster nodes. And the SAP NetWeaver, SAP Solutions and High Availability repos must be enabled on each cluster node.

Follow this [kbase article](#) to enable the repos on both nodes which are required for this environment.

### 2.2. PACEMAKER RESOURCE AGENTS

For a pacemaker-based HA cluster to manage both SAP HANA System Replication and also ENSA2 the following resource agents are required.

#### 2.2.1. SAPInstance

The **SAPInstance** resource agent will be used for managing the ASCS and ERS resources in this example. All operations of the **SAPInstance** resource agent are done by using the SAP start-up service framework **sapstartsrv**.

#### 2.2.2. SAPHanaTopology (Cloned Resource)

This resource agent is gathering status and configuration of SAP HANA System Replication on each cluster node. It is essential to have the data from this agent present in cluster node attributes for SAPHana resource agent to work properly.

#### 2.2.3. SAPHana (Promotable Cloned resource)

This resource is responsible for starting, stopping, and relocating (failover) of SAP HANA database. This resource agent takes information gathered by SAPHanaTopology and based on that it is interacting with SAP HANA database to do things. It also adds additional information to cluster node attributes about SAP HANA status on cluster nodes.

#### 2.2.4. filesystem

Pacemaker cluster resource-agent for Filesystem. It manages a Filesystem on a shared storage medium exported by NFS or iSCSI etc.

#### 2.2.5. IPaddr2 (or other RAs for managing VIPs on CCSPs)

Manages virtual IPv4 and IPv6 addresses and aliases.

### 2.3. TWO NODE CLUSTER ENVIRONMENT

Since this is a Cost-Optimized scenario, we will focus only on a 2-node cluster environment. ENSA1 can only be configured in a 2-node cluster where the ASCS can failover to the other node where the ERS is running. ENSA2, on the other hand, supports running more than 2 nodes in a cluster; however, SAP

HANA scale-up instances are limited to 2-node clusters only, therefore, this Cost-Optimized document keeps everything simple by using only 2 nodes in the cluster.

## 2.4. STORAGE REQUIREMENTS

Directories created for S/4HANA installation should be put on shared storage, following the below-mentioned rules:

### 2.4.1. Instance Specific Directory

There must be a separate SAN LUN or NFS export for the ASCS and ERS instances that can be mounted by the cluster on each node.

For example, as shown below, 'ASCS' and 'ERS' instances, respectively, the instance specific directory must be present on the corresponding node.

- ASCS node: **/usr/sap/SID/ASCS<Ins#>**
- ERS node: **/usr/sap/SID/ERS<Ins#>**
- Both nodes: **/hana/**
  - **Note:** As there will be System Replication, the `/hana/` directory is local that is non-shared on each node.

**Note:** For the Application Servers, the following directory must be made available on the nodes where the Application Server instances will run:

- App Server Node(s) (D<Ins#>): **/usr/sap/SID/D<Ins#>**

When using SAN LUNs for the instance directories, customers must use [HA-LVM](#) to ensure that the instance directories can only be mounted on one node at a time.

When using NFS exports, if the directories are created on the same directory tree on an NFS file server, such as Azure NetApp Files or Amazon EFS, the option `force_unmount=safe` must be used when configuring the Filesystem resource. This option will ensure that the cluster only stops the processes running on the specific NFS export instead of stopping all processes running on the directory tree where the exports are created.

### 2.4.2. Shared Directories

The following mount points must be available on ASCS, ERS, HANA, and Application Servers nodes.

**/sapmnt**  
**/usr/sap/trans**  
**/usr/sap/SID/SYS**

Shared storage can be achieved by:

- using the external NFS server (NFS server cannot run on any of the nodes inside the cluster in which the shares would be mounted. More details about this limitation can be found in the article [Hangs occur if a Red Hat Enterprise Linux system is used as both NFS server and NFS client for the same mount](#))
- using the GFS2 filesystem (this requires all nodes to have **Resilient Storage Add-on**)

- using the glusterfs filesystem (check the additional notes in the article [Can glusterfs be used for the SAP NetWeaver shared filesystems?](#))

These mount points must be either managed by the cluster or **mounted** before the cluster is started.

## CHAPTER 3. INSTALL SAP S/4HANA

### 3.1. CONFIGURATION OPTIONS USED IN THIS DOCUMENT

Below are the configuration options that will be used for instances in this document.

Both nodes will be running the ASCS/ERS and HDB instances with Automated System Replication in a cluster:

1st node hostname: s4node1  
2nd node hostname: s4node2

SID: S4H

ASCS Instance number: 20  
ASCS virtual hostname: s4ascs

ERS Instance number: 29  
ERS virtual hostname: s4ers

HANA database:

SID: S4D  
HANA Instance number: 00  
HANA virtual hostname: s4db

### 3.2. PREPARE HOSTS

Before starting installation ensure that you have to:

- Install RHEL 8 for SAP Solutions (the latest certified version for SAP HANA is recommended)
- Register the system to Red Hat Customer Portal or Satellite
- Enable RHEL for SAP Applications and RHEL for SAP Solutions repos
- Enable High Availability add-on channel
- Place shared storage and filesystems at correct mount points
- Make the virtual IP addresses, used by instances, present and reachable
- Resolve hostnames, that will be used by instances, to IP addresses and back
- Make available the installation media
- Configure the system according to the recommendation for running SAP S/4HANA
  - For more information please refer to Red Hat Enterprise Linux 8.x: Installation and Configuration - [SAP OSS Note 2772999](#)

### 3.3. INSTALL S/4HANA

Use SAP's Software Provisioning Manager (SWPM) to install instances in the following order:

- ASCS instance
- ERS instance
- SAP HANA DB instances on both nodes with System Replication

### 3.3.1. Install S/4 on s4node1

The following file systems should be mounted on s4node1, where ASCS will be installed:

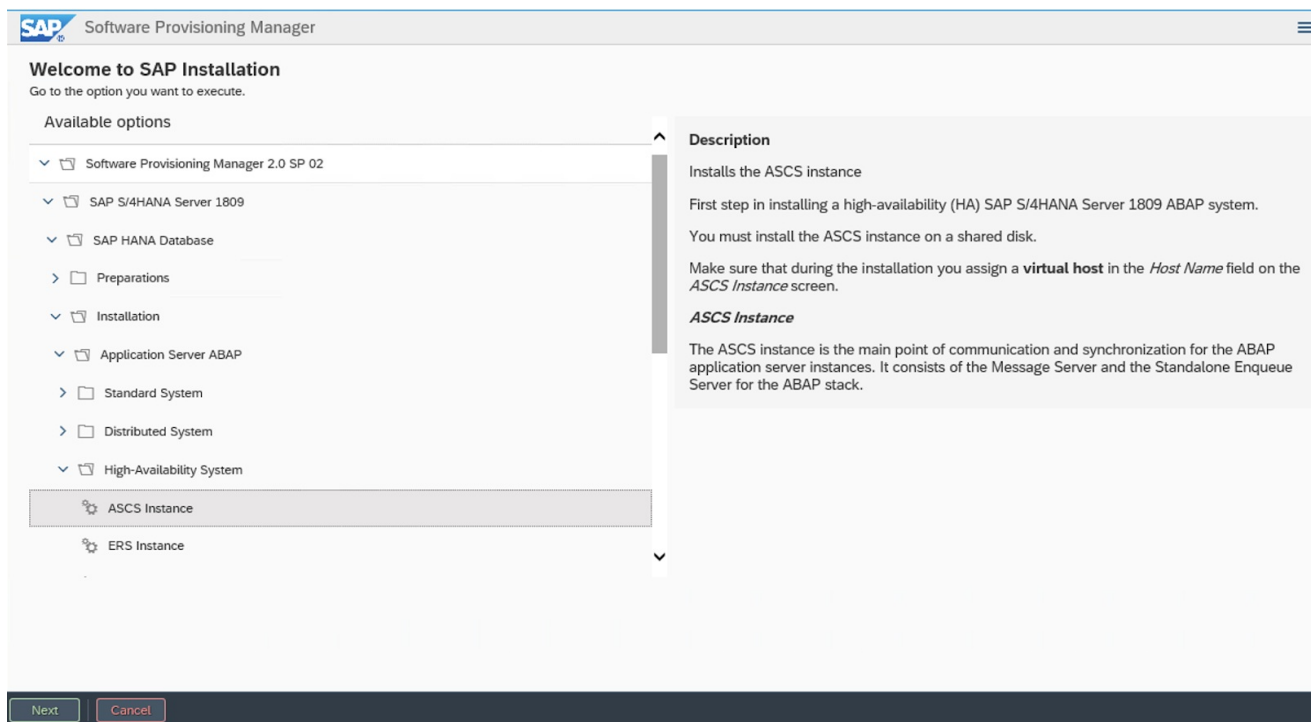
```
/usr/sap/S4H/ASCS20
/usr/sap/S4H/SYS
/usr/sap/trans
/sapmnt
```

Virtual IP for **s4ascs** should be enabled on s4node1.

Run the installer:

```
[root@s4node1]# ./sapinst SAPINST_USE_HOSTNAME=s4ascs
```

Select **High-Availability System** option.



### 3.3.2. Install ERS on s4node2

The following file systems should be mounted on s4node2, where ERS will be installed:

```
/usr/sap/S4H/ERS29 /usr/sap/S4H/SYS /usr/sap/trans /sapmnt
```

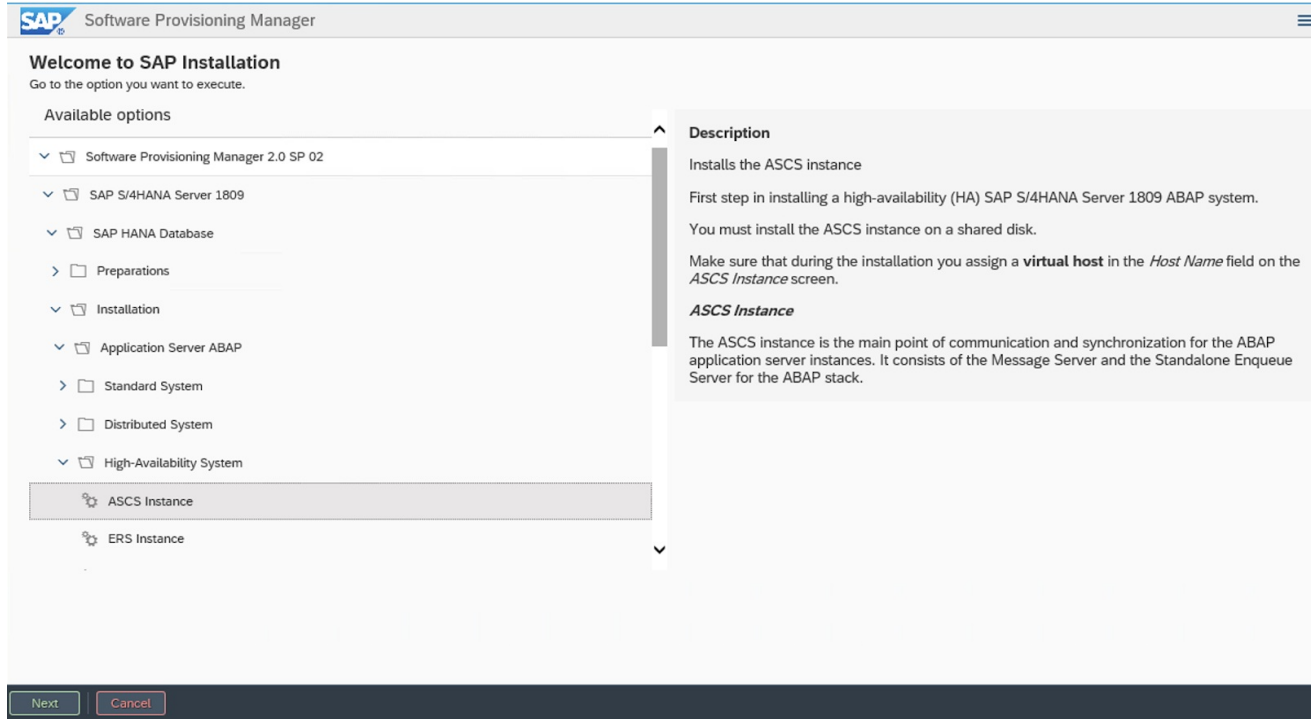


Virtual IP for **s4ers** should be enabled on s4node2.

Run the installer:

```
[root@s4node2]# ./sapinst SAPINST_USE_HOSTNAME=s4ers
```

Select **High-Availability System** option.



### 3.3.3. SAP HANA

In this example, we will be using SAP HANA with the following configuration. You can also use other supported databases as per the [support policies](#).

```
SAP HANA SID: S4D
SAP HANA Instance number: 00
```

In this example, SAP HANA Database server can be installed on both nodes using the hdblcm command line tool and then Automated HANA System Replication will be established in the same way as mentioned in the following document: [SAP HANA system replication in pacemaker cluster](#).

It is obvious to note that in this setup, the ASCS and HANA primary instances can failover independently of each other; hence, there will come a situation where the ASCS and the primary SAP HANA instances start running on the same node. Therefore, it is also important to ensure that both nodes have enough resources and free memory available for the ASCS/ERS and the primary SAP HANA instances to run on one node in an event of the other node's failure.

To achieve this, the SAP HANA instance can be set with certain memory restrictions/limits, and we highly recommend reaching out to SAP for exploring the available options as per your SAP HANA environment. Some related links are as follows:

- [Change the Global Memory Allocation Limit](#)

- [HANA: How to set allocation limit for tenant databases](#)

## 3.4. POST INSTALLATION

### 3.4.1. ASCS profile modification

The ASCS instance requires the following modification to its profile to prevent the automatic restart of the server instance since it will be managed by the cluster. To apply the change, run the following command at your ASCS profile **/sapmnt/S4H/profile/S4H\_ASCS20\_s4ascs**.

```
[root]# sed -i -e 's/Restart_Program_01/Start_Program_01/'
/sapmnt/S4H/profile/S4H_ASCS20_s4ascs+
```

### 3.4.2. ERS profile modification

The ERS instance requires the following modification to its profile, to prevent the automatic restart of the enqueue server, as it will be managed by cluster. To apply the change, run the following command at your ERS profile **/sapmnt/S4H/profile/S4H\_ERS29\_s4ers**.

```
[root]# sed -i -e 's/Restart_Program_00/Start_Program_00/'
/sapmnt/S4H/profile/S4H_ERS29_s4ers+
```

### 3.4.3. Update the /usr/sap/sapservices file

On both s4node1 and s4node2, make sure the following two lines are commented out in **/usr/sap/sapservices** file:

```
#LD_LIBRARY_PATH=/usr/sap/S4H/ERS29/exe:$LD_LIBRARY_PATH; export
LD_LIBRARY_PATH; /usr/sap/S4H/ERS29/exe/sapstartsrv
pf=/usr/sap/S4H/SYS/profile/S4H_ERS29_s4ers -D -u s4hadm
#LD_LIBRARY_PATH=/usr/sap/S4H/ASCS20/exe:$LD_LIBRARY_PATH; export
LD_LIBRARY_PATH; /usr/sap/S4H/ASCS20/exe/sapstartsrv
pf=/usr/sap/S4H/SYS/profile/S4H_ASCS20_s4ascs -D -u s4hadm
```

### 3.4.4. Create mount points for ASCS and ERS on the failover node

```
[root@s4node1 ~]# mkdir /usr/sap/S4H/ERS29/
[root@s4node1 ~]# chown s4hadm:sapsys /usr/sap/S4H/ERS29/

[root@s4node2 ~]# mkdir /usr/sap/S4H/ASCS20
[root@s4node2 ~]# chown s4hadm:sapsys /usr/sap/S4H/ASCS20
```

### 3.4.5. Manually Testing Instances on Other Node

Stop ASCS and ERS instances. Move the instance specific directory to the other node:

```
[root@s4node1 ~]# umount /usr/sap/S4H/ASCS20
[root@s4node2 ~]# mount /usr/sap/S4H/ASCS20
```

```
[root@s4node2 ~]# umount /usr/sap/S4H/ERS29/  
[root@s4node1 ~]# mount /usr/sap/S4H/ERS29/
```

Manually start the ASCS and ERS instances on the other cluster node, then manually stop them, respectively.

### 3.4.6. Check SAP HostAgent on all nodes

On all nodes check if SAP HostAgent has the same version and meets the minimum version requirement:

```
[root]# /usr/sap/hostctrl/exe/saphostexec -version
```

To upgrade/install SAP HostAgent. Please follow [SAP OSS Note 1031096](#).

### 3.4.7. Install permanent SAP license keys

SAP hardware key determination in the high-availability scenario has been improved. It might be necessary to install several SAP license keys based on the hardware key of each cluster node. Please see SAP OSS Note [1178686](#) - Linux: Alternative method to generate a SAP hardware key for more information.

## CHAPTER 4. INSTALL PACEMAKER

Please refer to the following documentation to first set up a pacemaker cluster.

- [Reference Document for the High Availability Add-On for Red Hat Enterprise Linux 7](#)
- [Configuring and Managing High Availability Clusters on RHEL 8](#)

Please make sure to follow the guidelines in [Support Policies for RHEL High Availability Clusters - General Requirements for Fencing/STONITH](#) for the fencing/STONITH setup. Information about the fencing/STONITH agents supported for different platforms are available at [Cluster Platforms and Architectures](#).

This guide will assume that the following things are working properly:

- Pacemaker cluster is configured according to documentation and has proper and working fencing
- Enqueue replication between the (A)SCS and ERS instances has been manually tested as explained in [Setting up Enqueue Replication Server fail over](#)
- The nodes are subscribed to the required channels as explained in [RHEL for SAP Repositories and How to Enable Them](#)

### 4.1. CONFIGURE GENERAL CLUSTER PROPERTIES

To avoid unnecessary failovers of the resources during initial testing and post production, set the following default values for the resource-stickiness and migration-threshold parameters. Note that defaults do not apply to resources which override them with their own defined values.

```
[root]# pcs resource defaults resource-stickiness=1
[root]# pcs resource defaults migration-threshold=3
```

**Warning:** As of RHEL 8.4 (pcs-0.10.8-1.el8), the commands above are deprecated. Use the commands below: `+[source,text]`

```
[root]# pcs resource defaults update resource-stickiness=1
[root]# pcs resource defaults update migration-threshold=3
```

#### Notes:

1. It is sufficient to run the commands above on one node of the cluster.
2. The command **resource-stickiness=1** will encourage the resource to stay running where it is, while **migration-threshold=3** will cause the resource to move to a new node after 3 failures. 3 is generally sufficient in preventing the resource from prematurely failing over to another node. This also ensures that the resource failover time stays within a controllable limit.

### 4.2. INSTALL RESOURCE-AGENTS-SAP ON ALL CLUSTER NODES

```
[root]# yum install resource-agents-sap
```

### 4.3. CONFIGURE CLUSTER RESOURCES FOR SHARED FILESYSTEMS

Configure shared filesystem to provide following mount points on all the cluster nodes.

```
/sapmnt
/usr/sap/trans
/usr/sap/S4H/SYS
```

### 4.3.1. Configure shared filesystems managed by the cluster

The cloned **Filesystem** cluster resource can be used to mount the shares from external NFS server on all cluster nodes as shown below.

```
[root]# pcs resource create s4h_fs_sapmnt Filesystem \
device='<NFS_Server>:<sapmnt_nfs_share>' directory='/sapmnt' \
fstype='nfs' --clone interleave=true
[root]# pcs resource create s4h_fs_sap_trans Filesystem \
device='<NFS_Server>:<sap_trans_nfs_share>' directory='/usr/sap/trans' \
fstype='nfs' --clone interleave=true
[root]# pcs resource create s4h_fs_sap_sys Filesystem \
device='<NFS_Server>:<s4h_sys_nfs_share>' directory='/usr/sap/S4H/SYS' \
fstype='nfs' --clone interleave=true
```

After creating the **Filesystem** resources verify that they have started properly on all nodes.

```
[root]# pcs status
...
Clone Set: s4h_fs_sapmnt-clone [s4h_fs_sapmnt]
Started: [ s4node1 s4node2 ]
Clone Set: s4h_fs_sap_trans-clone [s4h_fs_sap_trans]
Started: [ s4node1 s4node2 ]
Clone Set: s4h_fs_sys-clone [s4h_fs_sys]
Started: [ s4node1 s4node2 ]
...
```

### 4.3.2. Configure shared filesystems managed outside of cluster

In case that shared filesystems will NOT be managed by cluster, it must be ensured that they are available before the **pacemaker** service is started.

In RHEL 7 due to systemd parallelization you must ensure that shared filesystems are started in **resource-agents-deps** target. More details on this can be found in documentation section [9.6. Configuring Startup Order for Resource Dependencies not Managed by Pacemaker \(Red Hat Enterprise Linux 7.4 and later\)](#).

## 4.4. CONFIGURE ASCS RESOURCE GROUP

### 4.4.1. Create resource for virtual IP address

```
[root]# pcs resource create s4h_vip_ascs20 IPAddr2 ip=192.168.200.201 \
--group s4h_ASCS20_group
```

### 4.4.2. Create resource for ASCS filesystem.

Below is the example of creating resource for NFS filesystem

```
[root]# pcs resource create s4h_fs_ascs20 Filesystem \
```

```
device='<NFS_Server>:<s4h_asc20_nfs_share>' \
directory=/usr/sap/S4H/ASCS20 fstype=nfs force_unmount=safe \
--group s4h_ASCS20_group op start interval=0 timeout=60 \
op stop interval=0 timeout=120 \
op monitor interval=200 timeout=40
```

Below is the example of creating resources for HA-LVM filesystem

```
[root]# pcs resource create s4h_fs_asc20_lvm LVM \
volgrpname='<asc20_volume_group>' exclusive=true \
--group s4h_ASCS20_group

[root]# pcs resource create s4h_fs_asc20 Filesystem \
device='/dev/mapper/<asc20_logical_volumes>' \
directory=/usr/sap/S4H/ASCS20 fstype=ext4 \
--group s4h_ASCS20_group
```

### 4.4.3. Create resource for ASCS instance

```
[root]# pcs resource create s4h_asc20 SAPInstance \
InstanceName="S4H_ASCS20_s4asc20" \
START_PROFILE=/sapmnt/S4H/profile/S4H_ASCS20_s4asc20 \
AUTOMATIC_RECOVER=false \
meta resource-stickiness=5000 \
--group s4h_ASCS20_group \
op monitor interval=20 on-fail=restart timeout=60 \
op start interval=0 timeout=600 \
op stop interval=0 timeout=600
```

**Note:** `meta resource-stickiness=5000` is here to balance out the failover constraint with ERS so the resource stays on the node where it started and doesn't migrate around the cluster uncontrollably. Add a resource stickiness to the group to ensure that the ASCS will stay on a node if possible:

```
[root]# pcs resource meta s4h_ASCS20_group resource-stickiness=3000
```

## 4.5. CONFIGURE ERS RESOURCE GROUP

### 4.5.1. Create resource for virtual IP address

```
[root]# pcs resource create s4h_vip_ers29 IPAddr2 ip=192.168.200.202 \
--group s4h_ERS29_group
```

### 4.5.2. Create resource for ERS filesystem

Below is the example of creating resource for NFS filesystem

```
[root]# pcs resource create s4h_fs_ers29 Filesystem \
device='<NFS_Server>:<s4h_ers29_nfs_share>' \
directory=/usr/sap/S4H/ERS29 fstype=nfs force_unmount=safe \
--group s4h_ERS29_group op start interval=0 timeout=60 \
op stop interval=0 timeout=120 op monitor interval=200 timeout=40
```

Below is the example of creating resources for HA-LVM filesystem

```
[root]# pcs resource create s4h_fs_ers29_lvm LVM \
volgrpname='<ers_volume_group>' exclusive=true --group s4h_ERS29_group

[root]# pcs resource create s4h_fs_ers29 Filesystem \
device='/dev/mapper/<ers_logical_volume>' directory='/usr/sap/S4H/ERS29 \
fstype=ext4 --group s4h_ERS29_group
```

### 4.5.3. Create resource for ERS instance

Create an ERS instance cluster resource.

**Note:** In ENSA2 deployments the **IS\_ERS** attribute is optional. To learn more about **IS\_ERS**, additional information can be found in [How does the IS\\_ERS attribute work on a SAP NetWeaver cluster with Standalone Enqueue Server \(ENSA1 and ENSA2\)?](#).

```
[root]# pcs resource create s4h_ers29 SAPInstance \
InstanceName="S4H_ERS29_s4ers" \
START_PROFILE=/sapmnt/S4H/profile/S4H_ERS29_s4ers \
AUTOMATIC_RECOVER=false \
--group s4h_ERS29_group \
op monitor interval=20 on-fail=restart timeout=60 \
op start interval=0 timeout=600 \
op stop interval=0 timeout=600
```

## 4.6. CREATE CONSTRAINTS

### 4.6.1. Create colocation constraint for ASCS and ERS resource groups

Resource groups **s4h\_ASCS20\_group** and **s4h\_ERS29\_group** should try to avoid running on the same node. Order of groups matters.

```
[root]# pcs constraint colocation add s4h_ERS29_group with s4h_ASCS20_group \
-5000
```

### 4.6.2. Create location constraint for ASCS resource

ASCS20 instance **rh2\_ascs20** prefers to run on node where ERS was running before failover.

```
# pcs constraint location rh2_ascs20 rule score=2000 runs_ers_RH2 eq 1
```

### 4.6.3. Create order constraint for ASCS and ERS resource groups

Prefer to start **s4h\_ASCS20\_group** before the **s4h\_ERS29\_group**

```
[root]# pcs constraint order start s4h_ASCS20_group then start \
s4h_ERS29_group symmetrical=false kind=Optional
[root]# pcs constraint order start s4h_ASCS20_group then stop \
s4h_ERS29_group symmetrical=false kind=Optional
```

#### 4.6.4. Create order constraint for /sapmnt resource managed by cluster

If the shared filesystem **/sapmnt** is managed by the cluster, then the following constraints ensure that resource groups with ASCS and ERS SAPInstance resources are started only once the filesystem is available.

```
[root]# pcs constraint order s4h_fs_sapmnt-clone then s4h_ASCS20_group  
[root]# pcs constraint order s4h_fs_sapmnt-clone then s4h_ERS29_group
```



## CHAPTER 5. TEST THE CLUSTER CONFIGURATION

### 5.1. CHECK THE CONSTRAINTS

```
[root]# pcs constraint
Location Constraints:
Ordering Constraints:
start s4h_ASCS20_group then start s4h_ERS29_group (kind:Optional) (non-symmetrical)
start s4h_ASCS20_group then stop s4h_ERS29_group (kind:Optional) (non-symmetrical)
Colocation Constraints:
s4h_ERS29_group with s4h_ASCS20_group (score:-5000) Ticket Constraints:
```

### 5.2. FAILOVER ASCS DUE TO NODE CRASH

Before the crash, ASCS is running on s4node1 while ERS is running on s4node2.

```
[root@s4node1]# pcs status
...
Resource Group: s4h_ASCS20_group
s4h_fs_ascs20 (ocf::heartbeat:Filesystem): Started s4node1
s4h_vip_ascs20 (ocf::heartbeat:IPAddr2): Started s4node1
s4h_ascs20 (ocf::heartbeat:SAPInstance): Started s4node1
Resource Group: s4h_ERS29_group
s4h_fs_ers29 (ocf::heartbeat:Filesystem): Started s4node2
s4h_vip_ers29 (ocf::heartbeat:IPAddr2): Started s4node2
s4h_ers29 (ocf::heartbeat:SAPInstance): Started s4node2 ...
```

On s4node2, run the following command to monitor the status changes in the cluster:

```
[root@s4node2 ~]# crm_mon -Arf
```

Crash s4node1 by running the following command. Please note that connection to s4node1 will be lost after the command.

```
[root@s4node1 ~]# echo c > /proc/sysrq-trigger
```

On s4node2, monitor the failover process. After failover, cluster should be in such state, with ASCS and ERS both on s4node2.

```
[root@s4node2 ~]# pcs status
...
Resource Group: s4h_ASCS20_group
s4h_fs_ascs20 (ocf::heartbeat:Filesystem): Started
s4node2 s4h_vip_ascs20 (ocf::heartbeat:IPAddr2): Started s4node2
s4h_ascs20 (ocf::heartbeat:SAPInstance): Started s4node2
Resource Group: s4h_ERS29_group
s4h_fs_ers29 (ocf::heartbeat:Filesystem): Started
s4node2 s4h_vip_ers29 (ocf::heartbeat:IPAddr2): Started s4node2
s4h_ers29 (ocf::heartbeat:SAPInstance): Started s4node2
...
```

### 5.3. ERS MOVES TO THE PREVIOUSLY FAILED NODE

Bring s4node1 back online and start the cluster:

```
[root@s4node1 ~]# pcs cluster start
```

ERS should move to s4node1, while ASCS remains on s4node2. Wait for ERS to finish the migration, and at the end the cluster should be in such state:

```
[root@node1 ~]# pcs status
...
Resource Group: s4h_ASCS20_group
s4h_fs_ascs20 (ocf::heartbeat:Filesystem): Started
s4node2 s4h_vip_ascs20 (ocf::heartbeat:IPaddr2): Started s4node2
s4h_ascs20 (ocf::heartbeat:SAPInstance): Started s4node2
Resource Group: s4h_ERS29_group
s4h_fs_ers29 (ocf::heartbeat:Filesystem): Started
s4node1 s4h_vip_ers29 (ocf::heartbeat:IPaddr2): Started s4node1
s4h_ers29 (ocf::heartbeat:SAPInstance): Started s4node1
...
```

## CHAPTER 6. ENABLE CLUSTER TO AUTO-START AFTER REBOOT

The cluster is not yet enabled to auto-start after reboot. System admin needs to manually start the cluster after the node is fenced and rebooted.

After testing the previous section, when everything works fine, enable the cluster to auto-start after reboot:

```
[root@s4node1]# pcs cluster enable --all
```

**Note:** In some situations it can be beneficial not to have the cluster auto-start after a node has been rebooted. For example, if there is an issue with a filesystem that is required by a cluster resource, and the filesystem needs to be repaired first before it can be used again, having the cluster auto-start can be a failure because the filesystem doesn't work. This can cause even more trouble.

Now please rerun the tests in the previous section to make sure that the cluster still works fine. Please note that in section 5.3, there is no need to run command **pcs cluster start** after a node is rebooted. Cluster should automatically start after reboot.

By this point you have successfully configured a two-node cluster for ENSA2. You can either continue with intensive testing to get ready for production or optionally add more nodes to the cluster.

## CHAPTER 7. TEST FAILOVER

### 7.1. FAILOVER ASCS DUE TO NODE CRASH

Before the crash, ASCS was running on s4node1 while ERS was running on s4node2. On s4node2, run the following command to monitor the status changes in the cluster:

```
[root@s4node2 ~]# crm_mon -Arf
```

Crash s4node1 by running the following command. Please note that connection to s4node1 will be lost after the command.

```
[root@s4node1 ~]# echo c > /proc/sysrq-trigger
```

On s4node2, monitor the failover process. After failover, cluster should be in such state, with ASCS running on s4node3, and ERS remaining on s4node2.

```
[root@s4node2 ~]# pcs status
...
Resource Group: s4h_ASCS20_group
  s4h_fs_ascs20 (ocf::heartbeat:Filesystem): Started s4node1
  s4h_vip_ascs20 (ocf::heartbeat:IPaddr2): Started s4node1
  s4h_ascs20 (ocf::heartbeat:SAPInstance): Started s4node1
Resource Group: s4h_ERS29_group
  s4h_fs_ers29 (ocf::heartbeat:Filesystem): Started s4node2
  s4h_vip_ers29 (ocf::heartbeat:IPaddr2): Started s4node2
  s4h_ers29 (ocf::heartbeat:SAPInstance): Started s4node2
...
```

### 7.2. ERS REMAINS ON CURRENT NODE

Bring s4node1 back online. ERS should remain on the current node instead of moving back to s4node1.

### 7.3. TEST ERS CRASH

Similarly, test crash the node where ERS is running. The ERS group should failover to the spare node while ASCS remains intact on its current node. After the crashed node is back, the ERS group should not move back.

### 7.4. YOUR COST-OPTIMIZED SAP S/4HANA CLUSTER ENVIRONMENT CAN LOOK LIKE THE ONE BELOW

```
[root@s4node1 ~]# pcs status
Cluster name: SAP-S4-HANA
....
Node List:
 * Online: [ s4node1 s4node2 ]
....
Full List of Resources:
 * s4-fence (stonith:fence_rhevm): Started s4node1
 * Clone Set: fs_sapmnt-clone [fs_sapmnt]:
```

```
* Started: [ s4node1 s4node2 ]
* Clone Set: fs_sap_trans-clone [fs_sap_trans]:
* Started: [ s4node1 s4node2 ]
* Clone Set: fs_sap_SYS-clone [fs_sap_SYS]:
* Started: [ s4node1 s4node2 ]
* Resource Group: s4h_ASCS20_group:
* s4h_lvm_ascs20 (ocf::heartbeat:LVM-activate): Started s4node1
* s4h_fs_ascs20 (ocf::heartbeat:Filesystem): Started s4node1
* s4h_ascs20 (ocf::heartbeat:SAPInstance): Started s4node1
* s4h_vip_ascs20 (ocf::heartbeat:IPAddr2): Started s4node1
* Resource Group: s4h_ERS29_group:
* s4h_lvm_ers29 (ocf::heartbeat:LVM-activate): Started s4node2
* s4h_fs_ers29 (ocf::heartbeat:Filesystem): Started s4node2
* s4h_ers29 (ocf::heartbeat:SAPInstance): Started s4node2
* s4h_vip_ers29 (ocf::heartbeat:IPAddr2): Started s4node2
* Clone Set: SAPHanaTopology_S4D_00-clone [SAPHanaTopology_S4D_00]:
* Started: [ s4node1 s4node2 ]
* Clone Set: SAPHana_S4D_00-clone [SAPHana_S4D_00] (promotable):
  * Masters: [ s4node2 ]
* Slaves: [ s4node1 ]
* vip_S4D_00 (ocf::heartbeat:IPAddr2): Started s4node2
```

## CHAPTER 8. OPTIONAL - ENABLE SAP HA INTERFACE FOR MANAGEMENT OF CLUSTER-CONTROLLED ASCS/ERS INSTANCES USING SAP MANAGEMENT TOOLS

When a system admin controls a SAP instance that is running inside the Pacemaker cluster, either manually or using tools such as SAP Management Console (MC/MMC), the change needs to be done through the HA interface that's provided by the HA cluster software. SAP Start Service **sapstartsrv** controls the SAP instances and needs to be configured to communicate with the pacemaker cluster software through the HA interface.

Please follow the kbase article to configure the **HAlib**: [How to enable the SAP HA Interface for SAP ABAP application server instances managed by the RHEL HA Add-On?](#)