



Red Hat Enterprise Linux for SAP Solutions 8

Configuring HA clusters to manage SAP
NetWeaver or SAP S/4HANA Application
server instances using the RHEL HA Add-On

Red Hat Enterprise Linux for SAP Solutions 8 Configuring HA clusters to manage SAP NetWeaver or SAP S/4HANA Application server instances using the RHEL HA Add-On

Legal Notice

Copyright © 2024 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[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] 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[®] 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

This guide outlines the process of configuring HA clusters to manage SAP NetWeaver or SAP S/4HANA Application server instances using the RHEL HA Add-On.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE	5
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION	6
CHAPTER 1. OVERVIEW	7
1.1. INTRODUCTION	7
1.2. AUDIENCE	7
1.3. CONCEPTS	7
1.3.1. SAP NetWeaver or S/4HANA High Availability	7
1.3.2. ENSA1 vs. ENSA2	8
1.3.2.1. Standalone Enqueue Server (ENSA1)	8
1.3.2.2. Standalone Enqueue Server 2 (ENSA2)	8
1.4. RESOURCE AGENTS	10
1.4.1. SAPInstance resource agent	10
1.4.1.1. Important SAPInstance resource agent parameters	11
1.4.2. SAPDatabase resource agent	13
1.4.2.1. Important SAPDatabase resource agent parameters	14
1.5. MULTI-SID SUPPORT (OPTIONAL)	15
1.5.1. Unique SID and Instance Number	15
1.5.2. Sizing	15
1.5.3. Installation	16
1.6. SUPPORT POLICIES	16
CHAPTER 2. REQUIREMENTS	17
2.1. SUBSCRIPTIONS AND REPOSITORIES	17
2.2. STORAGE REQUIREMENTS	17
2.2.1. Local directories	17
2.2.2. Instance Specific Directories	17
2.2.3. Shared Directories	18
2.3. FENCING/STONITH	18
2.4. QUORUM	19
CHAPTER 3. INSTALLING SAP APPLICATION SERVER INSTANCES	20
3.1. CONFIGURATION OPTIONS USED IN THIS DOCUMENT	20
3.2. PREPARING THE CLUSTER NODES FOR INSTALLATION OF THE SAP INSTANCES	20
3.3. INSTALLING SAP INSTANCES	21
3.3.1. Installing (A)SCS on node1	21
3.3.2. Installing ERS on node2	22
3.3.3. Installing primary/additional application server instances	23
3.4. POST INSTALLATION	23
3.4.1. (A)SCS profile modification	24
3.4.2. ERS profile modification	24
3.4.3. Updating the /usr/sap/sapservices file	24
3.4.4. Creating mount points for the instance specific directories on the failover node	24
3.4.5. Verifying that the SAP instances can be started and stopped on all cluster nodes	25
3.4.6. Verifying that the correct version of SAP Host Agent is installed on all HA cluster nodes	25
3.4.7. Installing permanent SAP license keys	25
3.4.8. Additional changes required when using systemd enabled SAP instances	25
CHAPTER 4. SETTING UP THE CLUSTER	26
4.1. PERFORMING THE BASIC CLUSTER INSTALLATION ON EACH NODE	26
4.2. CONFIGURING GENERAL CLUSTER PROPERTIES	26

4.3. INSTALLING THE RESOURCE-AGENTS-SAP PACKAGE ON ALL CLUSTER NODES	26
4.4. CONFIGURING ACCESS TO SHARED FILE SYSTEMS	26
4.4.1. Configuring shared file systems managed by the cluster	27
4.4.2. Configuring shared file systems managed outside of cluster	27
4.5. CONFIGURING THE (A)SCS RESOURCE GROUP	27
4.5.1. Creating resource for managing the virtual IP address of the (A)SCS instance	27
4.5.2. Creating resource for managing the (A)SCS instance directory	27
4.5.2.1. NFS	28
4.5.2.2. HA-LVM	28
4.5.3. Creating resource for managing the (A)SCS instance	28
4.6. CONFIGURING THE ERS RESOURCE GROUP	29
4.6.1. Creating resource for managing the virtual IP address of the ERS instance	29
4.6.2. Creating resource for managing the ERS instance directory	29
4.6.2.1. NFS	29
4.6.2.2. HA-LVM	29
4.6.3. Creating resource for managing the ERS instance	30
4.7. CREATING CONSTRAINTS	30
4.7.1. Creating colocation constraint for (A)SCS and ERS resource groups	30
4.7.2. Creating location constraint for (A)SCS resource (ENSA1 only)	30
4.7.3. Creating order constraint for (A)SCS and ERS resource groups	30
4.7.4. Creating order constraints for /sapmnt resource managed by cluster	31
4.8. CONFIGURING CLUSTER RESOURCE GROUP FOR MANAGING DATABASE INSTANCES (OPTIONAL)	31
4.8.1. Creating resource for managing the virtual IP address of the database instance	31
4.8.2. Creating resource for managing the directories used by the database instance	31
4.8.2.1. NFS	32
4.8.2.2. HA-LVM	32
4.8.3. Configuring SAPDatabase cluster resource	32
4.9. CONFIGURING PRIMARY/ADDITIONAL APPLICATION SERVER (PAS/AAS) RESOURCE GROUP (OPTIONAL)	32
4.9.1. Creating resource for managing the Virtual IP address (VIP) for the PAS/AAS instance	33
4.9.2. Creating resource for managing the filesystem for the PAS/AAS instance directory	33
4.9.2.1. NFS	33
4.9.2.2. HA-LVM	33
4.9.3. Configuring PAS/AAS SAPInstance cluster resource	34
4.9.4. Configuring constraints	34
4.9.4.1. Configuring order constraints for the PAS/AAS resource group(s)	34
4.9.4.1.1. Deployments with s4h_SAPDatabase_group	34
4.9.4.1.2. Deployments with SAP HANA with System Replication as database	34
4.9.4.2. Configuring colocation constraint for PAS and AAS SAPInstance cluster resources (Optional)	34
4.9.4.3. Creating order constraint for /sapmnt resource managed by cluster	35
4.10. STANDALONE ENQUEUE SERVER 2 (ENSA2) MULTI-NODE CLUSTER (OPTIONAL)	35
4.10.1. OS Configuration	35
4.10.2. Creating users and groups	35
4.10.3. Creating local directories and mount points for the shared file systems	35
4.10.4. Installing the RHEL HA Add-On and the resource agents for managing SAP instances	36
4.10.5. Adding the node to the cluster	36
4.10.6. Updating fencing/STONITH configuration to include the 3rd node	36
4.10.7. Updating ERS resource configuration	36
4.11. ENABLING THE SAP HA INTERFACE TO ALLOW SAP INSTANCES CONTROLLED BY THE CLUSTER TO BE MANAGED BY SAP MANAGEMENT TOOLS (OPTIONAL)	37
4.12. ENABLING CLUSTER TO AUTO-START AT BOOT (OPTIONAL)	37

CHAPTER 5. TESTING THE CLUSTER CONFIGURATION	38
5.1. MANUALLY MOVING ASCS INSTANCE USING PCS COMMAND	38
5.2. MANUALLY MOVING OF THE ASCS INSTANCE USING SAPCONTROL (WITH SAP HA INTERFACE ENABLED)	39
5.3. TESTING FAILURE OF THE ASCS INSTANCE	40
5.4. TESTING FAILURE OF THE ERS INSTANCE	42
5.5. FAILOVER OF ASCS INSTANCE DUE TO NODE CRASH	43
5.6. FAILURE OF ERS INSTANCE DUE TO NODE CRASH	44
5.7. FAILURE OF ASCS INSTANCE DUE TO NODE CRASH (ENSA2)	45
CHAPTER 6. MAINTENANCE PROCEDURES	47
6.1. UPDATE RHEL AND THE RHEL HA ADD-ON	47
CHAPTER 7. REFERENCES	48
7.1. RED HAT	48
7.2. SAP	48

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

SAP NetWeaver or SAP S/4HANA based systems play an important role in many business processes; thus, it is critical to ensure the continuous and reliable availability of those systems to the business. This can be achieved by using HA clustering for managing the instances of such SAP NetWeaver or SAP S/4HANA systems.

The underlying idea of HA clustering is a fairly simple one: not a single large machine bears all of the load and risk, but rather one or more machines automatically drop in as an instant full replacement for the service or the machine that has failed. In the best case, this replacement process causes no interruption to the systems' users.

1.2. AUDIENCE

Designing highly available solutions and implementing them based on SAP NetWeaver or SAP S/4HANA can be very complex, so deep knowledge about each layer of the infrastructure and every aspect of the deployment is needed to ensure reliable, repeatable, accurate, and quick automated actions.

This document is intended for SAP and Red Hat certified or trained administrators and consultants who already have experience setting up SAP NetWeaver or S/4HANA application server instances and HA clusters using the RHEL HA add-on or other clustering solutions. Access to both the [SAP Support Portal](#) and the [Red Hat Customer Portal](#) is required to be able to download software and additional documentation.

Red Hat Consulting is highly recommended to set up the cluster and customize the solution to meet customers' data center requirements, which are normally more complex than the solution presented in this document.

1.3. CONCEPTS

1.3.1. SAP NetWeaver or S/4HANA High Availability

A typical SAP NetWeaver or S/4HANA environment consists of three distinctive components:

- SAP **(A)SCS** instance
- SAP application server instances (Primary Application Server (PAS) and Additional Application Server (AAS) instances)
- Database instance

The **(A)SCS** instance and the database instance are single points of failure (SPOF); therefore, it is important to ensure they are protected by an HA solution to avoid data loss or corruption and unnecessary outages of the SAP system. For more information on SPOF, please refer to [Single point of failure](#).

For the application servers, the enqueue lock table that is managed by the enqueue server is the most critical component. To protect it, SAP has developed the "Enqueue Replication Server" (**ERS**), which maintains a backup copy of the enqueue lock table. While the **(A)SCS** is running on one server, the **ERS** always needs to maintain a copy of the current enqueue table on another server.

This document describes how to set up a two-node or three-node HA cluster solution for managing

(A)SCS and **ERS** instances that conforms to the guidelines for high availability that have been established by both SAP and Red Hat. The HA solution can either be used for the "Standalone Enqueue Server" (ENSA1) that is typically used with SAP NetWeaver or the "Standalone Enqueue Server 2" (ENSA2) that is used by SAP S/4HANA.

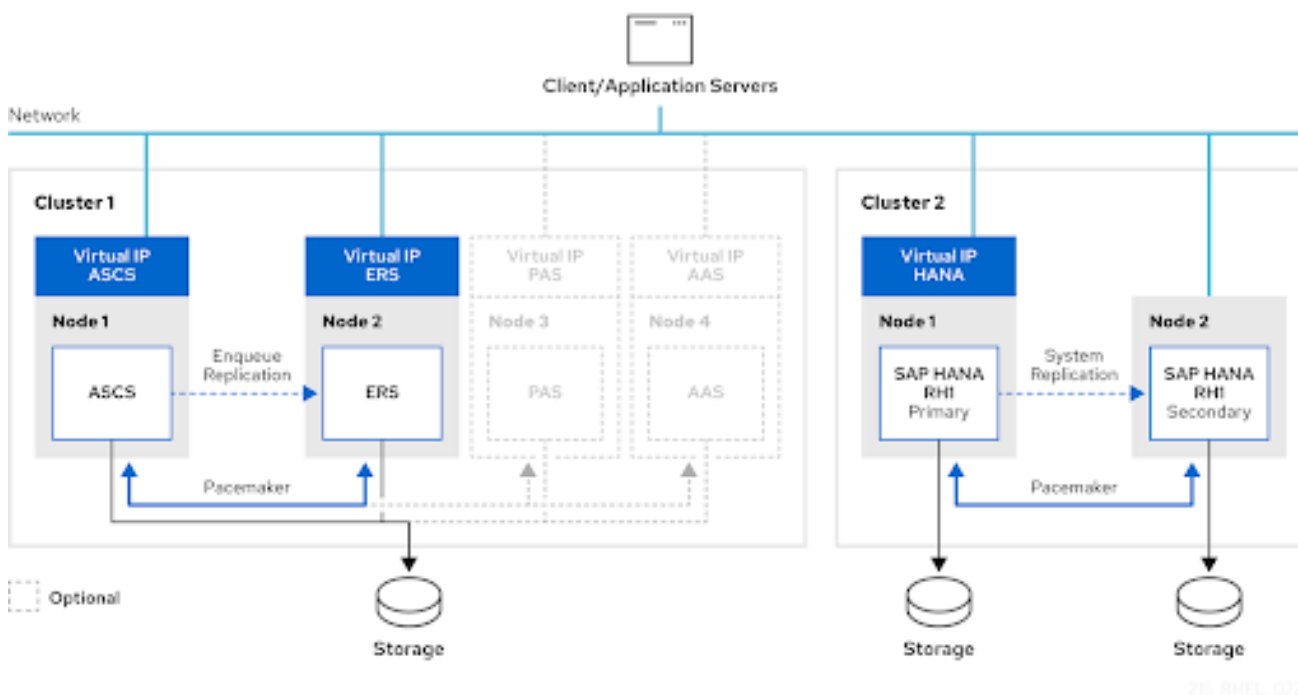
Additionally, it also provides guidelines for setting up HA cluster resources for managing other SAP instance types, like Primary Application Server (PAS) or Additional Application Server (AAS) instances that can either be managed as part of the same HA cluster or on a separate HA cluster.

1.3.2. ENSA1 vs. ENSA2

1.3.2.1. Standalone Enqueue Server (ENSA1)

In case there is an issue with the **(A)SCS** instance, for the Standalone Enqueue Server (ENSA1), it is required that the **(A)SCS** instance "follows" the **ERS** instance. That is, an HA cluster has to start the **(A)SCS** instance on the host where the **ERS** instance is currently running. Until the host where the **(A)SCS** instance was running has been fenced, it can be noticed that both instances stay running on that same node. When the HA cluster node where the **(A)SCS** instance was previously running is back online, the HA cluster should move the **ERS** instance to that HA cluster node so that Enqueue Replication can resume.

The following diagram shows the typical architecture of a Pacemaker HA cluster for managing SAP NetWeaver setups with the Standalone Enqueue Server (ENSA1).



Even though the diagram shows that it is optionally possible to also have Primary and Additional Application Server (PAS/AAS) instances managed on separate servers, it is also supported to have these instances running on the same HA cluster nodes as the **(A)SCS** and **ERS** instances and have them managed by the cluster.

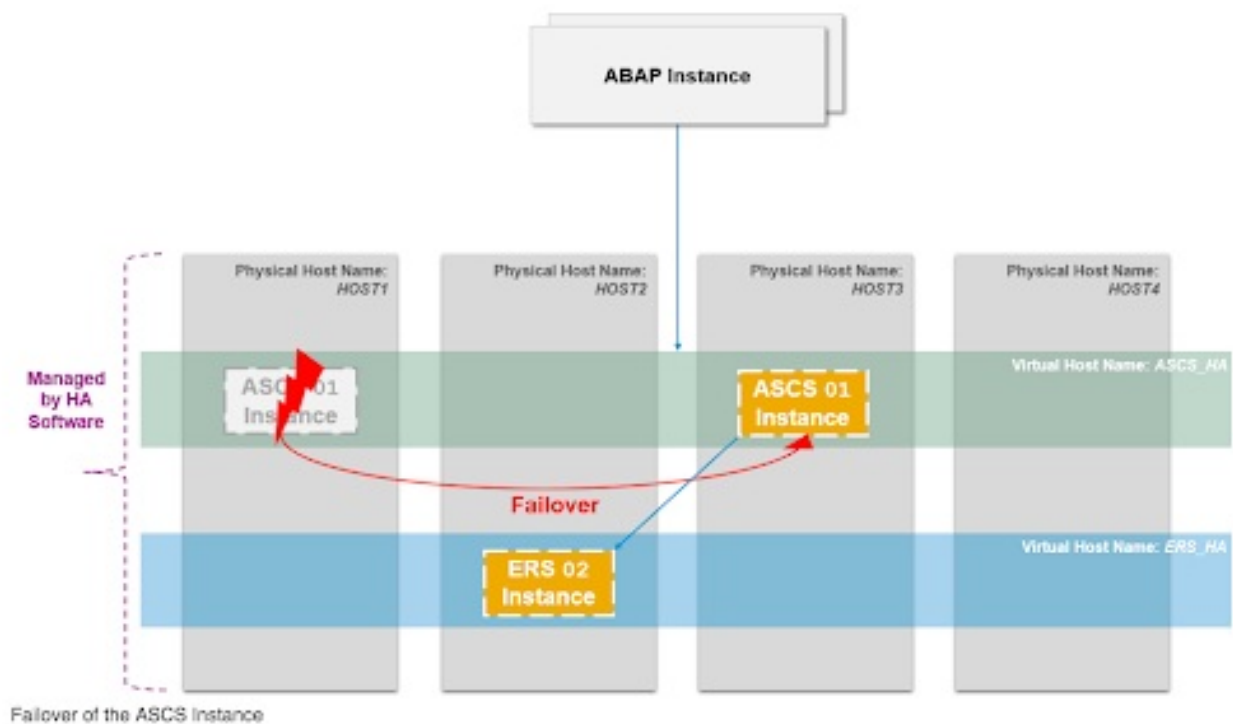
Please see the following SAP documentation for more information on how the Standalone Enqueue Server (ENSA1) works: [Standalone Enqueue Server](#).

1.3.2.2. Standalone Enqueue Server 2 (ENSA2)

As shown above with ENSA1, if there is a failover, the Standalone Enqueue Server is required to "follow" the Enqueue Replication Server. That is, the HA software had to start the **(A)SCS** instance on the host where the **ERS** instance is currently running.

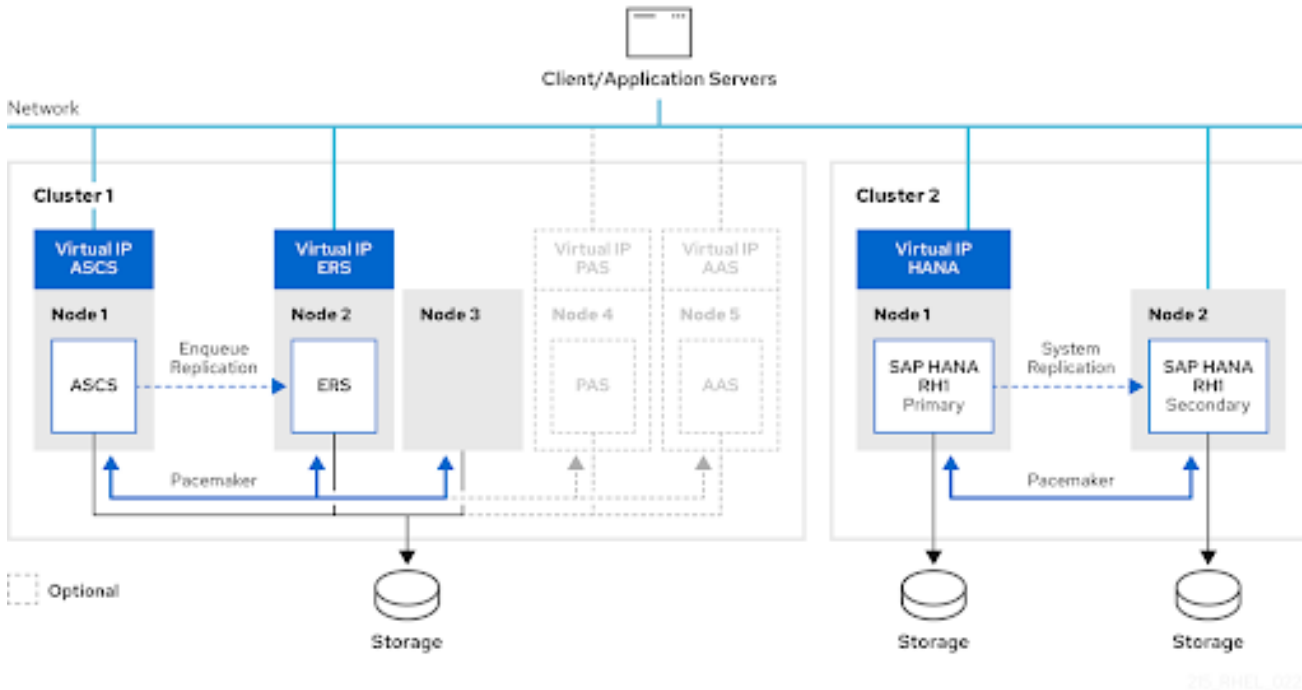
In contrast to the Standalone Enqueue Server (ENSA1), the new Standalone Enqueue Server 2 (ENSA2) and Enqueue Replicator 2 no longer have these restrictions, which means that the **ASCS** instance can either be restarted on the same cluster node in case of a failure. Or it can also be moved to another HA cluster node, which doesn't have to be the HA cluster node where the **ERS** instance is running. This makes it possible to use a multi-node HA cluster setup with more than two HA cluster nodes when Standalone Enqueue Server 2 (ENSA2) is used.

When using more than two HA cluster nodes, the **ASCS** will failover to a spare node, as illustrated in the following picture:



For more information on ENSA2, please refer to SAP Note [2630416 - Support for Standalone Enqueue Server 2](#).

The following diagram shows the architecture of a three-node cluster that can be used for managing SAP S/4HANA setups with the Standalone Enqueue Server 2 (ENSA2).



25_RHEL_022

Even though the diagram shows that it is optionally possible to also have Primary and Additional Application Server (PAS/AAS) instances managed on separate servers, it is also supported to have these instances running on the same HA cluster nodes as the **ASCS** and **ERS** instances and have them managed by the cluster.

For SAP S/4HANA, it is also possible to use a “cost-optimized” HA cluster setup, where the cluster nodes used for managing the HANA System Replication setup are also used for managing the **ASCS** and **ERS** instances. Please see [Configuring a Cost-Optimized SAP S/4HANA HA cluster \(HANA System Replication + ENSA2\) using the RHEL HA Add-On](#), for more information.

1.4. RESOURCE AGENTS

The following resource agents are provided on RHEL 8 for managing different instance types of SAP environments via the [resource-agents-sap RPM package](#).

1.4.1. SAPInstance resource agent

The **SAPInstance** resource agent can be used for managing SAP application server instances using the SAP Start Service that is part of the SAP Kernel. In addition to the (A)SCS, ERS, PAS, and AAS instances, it can also be used for managing other SAP instance types, like standalone SAP Web Dispatcher or standalone SAP Gateway instances (see [How to manage standalone SAP Web Dispatcher instances using the RHEL HA Add-On](#) for information on how to configure a pacemaker resource for managing such instances).

All operations of the **SAPInstance** resource agent are done by using commands provided by the SAP Startup Framework, which communicate with the **sapstartsrv** process of each SAP instance. **sapstartsrv** knows 4 status colors:

Color	Meaning
GREEN	Everything is fine.

Color	Meaning
YELLOW	Something is wrong, but the service is still working.
RED	The service does not work.
GRAY	The service has not been started.

The **SAPInstance** resource agent will interpret GREEN and YELLOW as OK, while statuses RED and GRAY are reported as NOT_RUNNING to the cluster.

The versions of the SAPInstance resource agent shipped with RHEL 8 also support SAP instances that are managed by the systemd-enabled SAP Startup Framework (see [The Systemd-Based SAP Startup Framework](#) for further details).

1.4.1.1. Important SAPInstance resource agent parameters

Attribute Name	Required	Default value	Description
InstanceName	yes	null	The full SAP instance profile name (<code><SAPSID>_<INSTANCE+INSTNO>_<virt hostname></code>), for example, S4H_ASCS20_s4asc s .
START_PROFILE	no	null	The full path to the SAP Start Profile (with SAP NetWeaver 7.1 and newer, the SAP Start profile is identical to the instance profile).

Attribute Name	Required	Default value	Description
IS_ERS	no	false	Only used for ASCS/ERS SAP Netweaver installations without implementing a promotable resource to allow the ASCS to find the ERS running on another cluster node after a resource failure. This parameter should be set to true for the resource used for managing the ERS instance for implementations following the SAP NetWeaver 7.50 HA certification (NW-HA-CLU-750; ENSA1). This also includes systems for NetWeaver less than 7.50 when using ENSA1.
DIR_EXECUTABLE	no	null	The full qualified path where to find sapstartsrv and sapcontrol binaries (only needed if the default location of the SAP Kernel binaries has been changed).
DIR_PROFILE	no	null	The full qualified path where to find the SAP START profile (only needed if the default location for the instance profiles has been changed).

Attribute Name	Required	Default value	Description
AUTOMATIC_RECOVER	no	false	The SAPInstance resource agent tries to recover a failed start attempt automatically one time. This is done by killing running instance processes, removing the kill.sap file, and executing cleanipc . Sometimes a crashed SAP instance leaves some processes and/or shared memory segments behind. Setting this option to true will try to remove those leftovers during a start operation.
MONITOR_SERVICES	no	disp+work msg_server enserver enrepserver jcontrol jstart	The list of services of an SAP instance that need to be monitored to determine the health of the instance. To monitor more/less, or other services that sapstartsrv supports, the list can be changed using this parameter. Names must match the strings used in the output of the command sapcontrol -nr [Instance-Nr] -function GetProcessList and multiple services separated by a (pipe) sign can be specified (the value for this parameter must always be the full list of services to monitor).

The full list of parameters can be obtained by running **pcs resource describe SAPInstance**.

1.4.2. SAPDatabase resource agent

The **SAPDatabase** resource agent can be used to manage single Oracle, IBM DB2, SAP ASE, or MaxDB database instances as part of a SAP NetWeaver based HA cluster setup. For more information, refer to [Support Policies for RHEL High Availability Clusters - Management of SAP NetWeaver in a Cluster](#) for

the list of supported database versions on RHEL 8.

The **SAPDatabase** resource agent does not run any database commands directly. It uses the **SAP Host Agent** to control the database. Therefore, the SAP Host Agent must be installed on each cluster node.

Since the **SAPDatabase** resource agent only provides basic functionality for managing database instances, it is recommended to use the HA features of the databases instead (for example, Oracle RAC and IBM DB2 HA/DR) if more HA capabilities are required for the database instance.

For S/4HANA HA setups, it is recommended to use HANA System Replication to make the HANA instance more robust against failures. The HANA System Replication HA setup can either be done using a separate cluster, or alternatively, it is also possible to use a “cost-optimized” S/4HANA HA setup where the **ASCS** and **ERS** instances are managed by the same HA cluster that is used for managing the HANA System Replication setup.

1.4.2.1. Important SAPDatabase resource agent parameters

Attribute Name	Required	Default value	Description
SID	yes	null	The unique database system identifier (usually identical to the SAP SID).
DBTYPE	yes	null	The type of database to manage. Valid values are: ADA (SAP MaxDB), DB6 (IBM DB2), ORA (Oracle DB), and SYB (SAP ASE).
DBINSTANCE	no	null	Must be used for special database implementations when the database instance name is not equal to the SID (e.g., Oracle DataGuard).
DBOSUSER	no	ADA=taken from /etc/opt/sdb , DB6= db2SID , ORA= oraSID and oracle , SYB= sybSID , HDB= SIDadm	The parameter can be set if the database processes on the operating system level are not executed with the default user of the used database type.

Attribute Name	Required	Default value	Description
STRICT_MONITORING	no	false	This controls how the resource agent monitors the database. If set to true , it will use saphostctrl -function GetDatabaseStatus to test the database state. If set to false , only operating system processes are monitored.
MONITOR_SERVICES	no	Instance Database Listener	Defines which services are monitored by the SAPDatabase resource agent if STRICT_MONITORING is set to true . Service names must correspond with the output of the saphostctrl -function GetDatabaseStatus command.
AUTOMATIC_RECOVER	no	false	If you set this to true , saphostctrl -function StartDatabase will always be called with the -force option.

The full list of parameters can be obtained by running **pcs resource describe SAPDatabase**.

1.5. MULTI-SID SUPPORT (OPTIONAL)

The setup described in this document can also be used to manage the **(A)SCS/ERS** instances for multiple SAP environments (Multi-SID) within the same HA cluster. For example, SAP products that contain both ABAP and Java application server instances (like SAP Solution Manager) could be candidates for a Multi-SID cluster.

However, some additional considerations need to be taken into account for such setups.

1.5.1. Unique SID and Instance Number

To avoid conflicts, each pair of **(A)SCS/ERS** instances must use a different SID, and each instance must use a unique Instance Number even if they belong to a different SID.

1.5.2. Sizing

Each HA cluster node must meet the SAP requirements for sizing to support multiple instances.

1.5.3. Installation

For each **(A)SCS/ERS** pair, please repeat all the steps documented in sections 4.5, 4.6, and 4.7. Each **(A)SCS/ERS** pair will failover independently, following the configuration rules.



NOTE

With the default pacemaker configuration for RHEL 8, certain failures of resource actions (for example, the stop of a resource fails) will cause the cluster node to be fenced. This means that, for example, if the stop of the resource for one **(A)SCS** instance on a HA cluster node fails, it would cause an outage for all other resources running on the same HA cluster node. Please see the description of the on-fail property for monitoring operations in [Configuring and managing high availability clusters - Chapter 21. Resource monitoring operations](#) for options on how to modify this behavior.

1.6. SUPPORT POLICIES

- [Support Policies for RHEL High Availability Clusters - Management of SAP S/4HANA](#)
- [Support Policies for RHEL High Availability Clusters - Management of SAP NetWeaver in a Cluster](#)

CHAPTER 2. REQUIREMENTS

2.1. SUBSCRIPTIONS AND REPOSITORIES

It is important to keep the subscription, kernel, and patch level identical on all cluster nodes and to ensure that the correct repositories are enabled.

Check out the following documentation for guidelines on how to enable the required subscriptions and repositories for running SAP NetWeaver or SAP S/4HANA application servers on RHEL 8 and have them managed by the RHEL HA Add-On: [RHEL for SAP Subscriptions and Repositories](#).

2.2. STORAGE REQUIREMENTS

The directories used by a SAP S/4HANA installation that is managed by the cluster must be set up according to the guidelines provided by SAP. See [SAP Directories](#) for more information.

2.2.1. Local directories

As per [SAP's guidance](#), the `/usr/sap/`, `/usr/sap/SYS/`, and `/usr/sap/<SAPSID>/` directories should be created locally on each node. While `/usr/sap/` will contain some additional files and directories after the installation of the SAP system that are specific to the node (for example, `/usr/sap/sapservices`, and `/usr/sap/hostctrl`), `/usr/sap/SYS/` only contains symlinks to other files and directories, and `/usr/sap/<SAPSID>/` is primarily used as a mountpoint for the instance-specific directories.

2.2.2. Instance Specific Directories

For the **(A)SCS**, **ERS**, and any other application server instance that is managed by the cluster, the instance-specific directory must be created on a separate SAN LUN or **NFS** export that can be mounted by the cluster as a local directory on the node where an instance is supposed to be running. For example:

- **(A)SCS:** `/usr/sap/<SAPSID>/ASCS<Ins#>/`
- **ERS:** `/usr/sap/<SAPSID>/ERS<Ins#>/`
- **App Server:** `/usr/sap/<SAPSID>/D<Ins#>/`

The cluster configuration must include resources for managing the filesystems for the instance directories as part of the resource group that is used to manage the instance and the virtual IP so that the cluster can automatically mount the filesystem on the node where the instance should be running.

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

The resources for managing the logical volumes (if SAN LUNS are used) and the filesystems must always be configured before the resource that is used for managing the SAP instance to ensure that the filesystem is mounted when the cluster attempts to start the instance itself.

With the exception of **NFS**, using a shared file system (for example, GFS2) to host all the instance-specific directories and make them available on all cluster nodes at the same time is not supported for the solution described in this document.

When using **NFS** exports for specific directories, if the directories are created on the same directory tree on an **NFS** file server, such as Azure NetApp Files (ANF) 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 have been created (see [During failover of a pacemaker resource, a Filesystem resource kills processes not using the filesystem](#) for more information).

2.2.3. Shared Directories

The following directories must be available on all servers running SAP instances of an SAP system:

- **/sapmnt/**
- **/usr/sap/trans/**

The **/sapmnt/** directory must also be accessible on all other servers that are running services that are part of the SAP system (for example, the servers hosting the HANA DB instances or servers hosting additional application servers not managed by the cluster).

To share the **/sapmnt/** and **/usr/sap/trans/** directories between all the servers hosting services of the same SAP system, either one of the following methods can be used:

- Using an external **NFS** server (as documented in [Support Policies for RHEL High Availability Clusters - Management of Highly Available Filesystem Mounts](#) using the same host as an **NFS** server and as an **NFS** client that mounts the same **NFS** exports ("loopback mounts") from this **NFS** server at the same time is not supported).
- Using the GFS2 filesystem (this requires all nodes to have [Resilient Storage Add-on](#) subscriptions, including servers that are running SAP instances not managed by the cluster).

The shared directories can either be statically mounted via **/etc/fstab** or the mounts can be managed by the cluster (in this case, it must be ensured that the cluster mounts the **/sapmnt/** directory on the cluster nodes before attempting to start any SAP instances by setting up appropriate constraints).

2.3. FENCING/STONITH

As documented at [Support Policies for RHEL High Availability Clusters - General Requirements for Fencing/STONITH](#), a working Fencing/STONITH device must be enabled on each cluster node in order for an HA cluster setup using the RHEL HA Add-on to be fully supported.

Which Fencing/STONITH device to use depends on the platform the cluster is running on. Please check out the [Fencing/STONITH](#) section in the [Support Policies for RHEL High Availability Clusters](#) for recommendations on fencing agents, or consult with your hardware or cloud provider to find out which fence device to use on their platform.



NOTE

Using **fence_scsi/fence_mpath** as the fencing device for HA cluster setups for managing SAP NetWeaver/S/4HANA application server instances is not a supported option since, as documented in [Support Policies for RHEL High Availability Clusters - fence_scsi and fence_mpath](#) these fence devices can only be used for cluster setups that manage shared storage, which is simultaneously accessed by multiple clients for reading and writing. Since the main purpose of a HA cluster for managing SAP NetWeaver/S/4HANA is to manage the SAP application server instances and not the shared directories that are needed in such environments, using **fence_scsi/fence_mpath** could result in the SAP instances not being stopped in case a node needs to be fenced (since **fence_scsi/fence_mpath** normally only block access to the storage devices managed by the cluster).

2.4. QUORUM

While pacemaker provides some built-in mechanisms to determine if a cluster is quorate or not, in some situations it might be desirable to add additional “quorum devices” in the cluster setup to help the cluster determine which side of the cluster should stay up and running in case a “split-brain” situation occurs.

For HA cluster setups that are used for managing SAP Application server instances, a quorum device is not required by default, but it is possible to add quorum devices to such setups if needed.

The options for setting up quorum devices vary depending on the configuration. Please review the following guidelines for more information:

- [Design Guidance for RHEL High Availability Clusters - Considerations with qdevice Quorum Arbitration.](#)

CHAPTER 3. INSTALLING SAP APPLICATION SERVER INSTANCES

3.1. CONFIGURATION OPTIONS USED IN THIS DOCUMENT

Below are the configuration options that will be used for instances in this document. Please adapt these options according to your local requirements.

For the HA cluster nodes and the **(A)SCS** and **ERS** instances managed by the HA cluster, the following values are used:

```
1st HA cluster node name:  node1
2nd HA cluster node name:  node2

SID:          S4H

ASCS Instance number:  20
ASCS virtual hostname: s4ascs
ASCS virtual IP address: 192.168.200.101

ERS Instance number:   29
ERS virtual hostname:  s4ers
ASCS virtual IP address: 192.168.200.102
```

For the optional primary application server (PAS) and additional application server (AAS) instances, the following values are used:

```
PAS Instance number:  21
PAS virtual hostname: s4pas
PAS virtual IP address: 192.168.200.103

AAS Instance number:  22
AAS virtual hostname: s4aas
AAS virtual IP address: 192.168.200.104
```

3.2. PREPARING THE CLUSTER NODES FOR INSTALLATION OF THE SAP INSTANCES

Before starting the installation, ensure that:

- RHEL 8 is installed and configured on all HA cluster nodes according to the recommendations from SAP and Red Hat for running SAP application server instances on RHEL 8.
- The RHEL for SAP Applications or RHEL for SAP Solutions subscriptions are activated, and the required repositories are enabled on all HA cluster nodes, as documented in [RHEL for SAP Subscriptions and Repositories](#).
- Shared storage and instance directories are present at the correct mount points.
- The virtual hostnames and IP addresses used by the SAP instances can be resolved in both directions, and the virtual IP addresses must be accessible.

- The SAP installation media are accessible on each HA cluster node where a SAP instance will be installed.

These setup steps can be partially automated using Ansible and **rhel-system-roles-sap system roles**. For more information on this, please check out [Red Hat Enterprise Linux System Roles for SAP](#) .

3.3. INSTALLING SAP INSTANCES

Using software provisioning manager (SWPM), install instances in the following order:

- **(A)SCS** instance
- **ERS** instance
- DB instance
- PAS instance
- AAS instances

The following sections just provide some specific recommendations that should be followed when installing SAP instances that will be managed by the HA cluster setup described in this document. Please check the official SAP installation guides for detailed instructions on how to install SAP NetWeaver or S/4HANA application server instances.

3.3.1. Installing (A)SCS on node1

The local directories and mount points required by the SAP instance must be created on the HA cluster node where the **(A)SCS** instance will be installed:

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

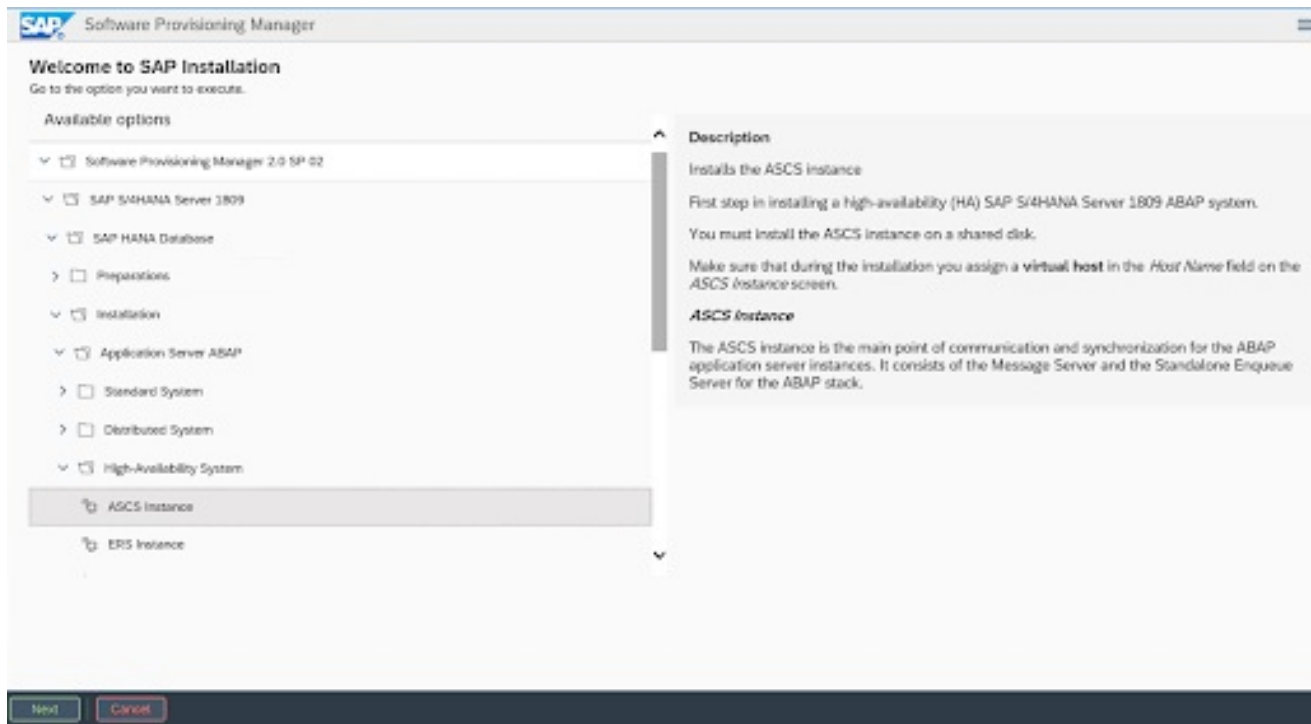
The shared directories and the instance directory must be manually mounted before starting the installation.

Also, the virtual IP address for the **(A)SCS** instance must be enabled on node 1, and it must have been verified that the virtual hostname for the **ERS** instance resolves to the virtual IP address.

When running the SAP installer, please make sure to specify the virtual hostname that should be used for the **(A)SCS** instance:

```
[root@node1]# ./sapinst SAPINST_USE_HOSTNAME=s4ascsc
```

Select the **High-Availability System** option for the installation of the **(A)SCS** instance:



3.3.2. Installing ERS on node2

The local directories and mount points required by the SAP instance must be created on the HA cluster node where the **ERS** instance will be installed:

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

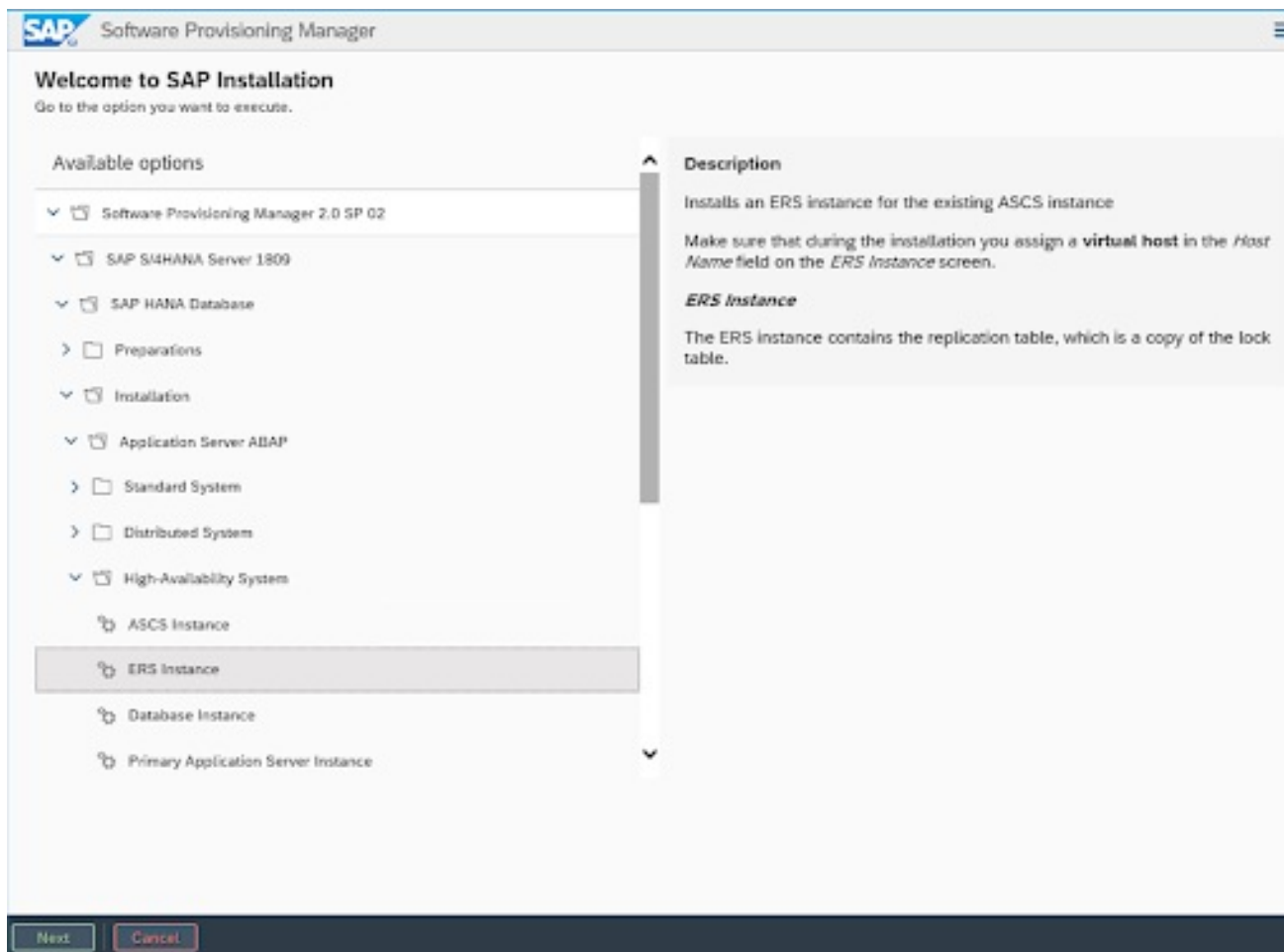
The shared directories and the instance directory must be manually mounted before starting the installation.

Also, the virtual IP address for the **ERS** instance must be enabled on node 2, and it must have been verified that the virtual hostname for the **ERS** instance resolves to the virtual IP address.

Make sure to specify the virtual hostname for the **ERS** instance when starting the installation:

```
[root@node2]# ./sapinst SAPINST_USE_HOSTNAME=s4ers
```

Select the High-Availability System option for the installation of the **ERS** instance:



3.3.3. Installing primary/additional application server instances

The local directories and mount points required by the SAP instance must be created on the HA cluster node where the primary or additional application server instance(s) will be installed:

```
/sapmnt/
/usr/sap/
/usr/sap/SYS/
/usr/sap/trans/
/usr/sap/S4H/
/usr/sap/S4H/D<Ins#>
```

The shared directories and the instance directory must be manually mounted before starting the installation.

Also, the virtual IP address for the application server instance must be enabled on the HA cluster node, and it must have been verified that the virtual hostname for the application server instance resolves to the virtual IP address.

Specify the virtual hostname for the instance when starting the installer:

```
[root@node<X>]# ./sapinst SAPINST_USE_HOSTNAME=<virtual hostname of instance>
```

Select the High-Availability System option for the installation of the application server instance.

3.4. POST INSTALLATION

3.4.1. (A)SCS profile modification

The **(A)SCS** instance profile has to be modified to prevent the automatic restart of the enqueue server process by the **sapstartsrv** process of the instance, since the instance will be managed by the cluster.

To modify the **(A)SCS** instance profile, run the following command:

```
[root@node1]# 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 profile has to be modified to prevent the automatic restart of the enqueue replication server process by the **sapstartsrv** of the instance since the **ERS** instance will be managed by the cluster.

To modify the **ERS** instance profile, run the following command:

```
[root@node2]# sed -i -e 's/Restart_Program_00/Start_Program_00/'
/sapmnt/S4H/profile/S4H_ERS29_s4ers
```

3.4.3. Updating the /usr/sap/sapservices file

To prevent the SAP instances that will be managed by the HA cluster to be started outside of the control of the HA cluster, make sure the following lines are commented out in the **/usr/sap/sapservices** file on all cluster nodes:

```
#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/ASC20/exe:$LD_LIBRARY_PATH; export
LD_LIBRARY_PATH; /usr/sap/S4H/ASC20/exe/sapstartsrv
pf=/usr/sap/S4H/SYS/profile/S4H_ASC20_s4ascs -D -u s4hadm

#LD_LIBRARY_PATH=/usr/sap/S4H/D21/exe:$LD_LIBRARY_PATH; export LD_LIBRARY_PATH;
/usr/sap/S4H/D21/exe/sapstartsrv pf=/usr/sap/S4H/SYS/profile/S4H_D21_s4hpas -D -u s4hadm

#LD_LIBRARY_PATH=/usr/sap/S4H/D22/exe:$LD_LIBRARY_PATH; export LD_LIBRARY_PATH;
/usr/sap/S4H/D22/exe/sapstartsrv pf=/usr/sap/S4H/SYS/profile/S4H_D22_s4haas -D -u s4hadm
```

3.4.4. Creating mount points for the instance specific directories on the failover node

The mount points where the instance-specific directories will be mounted have to be created and the user and group ownership must be set to the **<sid>adm** user and the sapsys group on all HA cluster nodes:

```
[root@node1]# mkdir /usr/sap/S4H/ERS29/
[root@node1]# chown s4hadm:sapsys /usr/sap/S4H/ERS29/

[root@node2]# mkdir /usr/sap/S4H/ASC20
```

```
[root@node2]# chown s4hadm:sapsys /usr/sap/S4H/ASCS20  
[root@node<x>]# mkdir /usr/sap/S4H/D<Ins#>  
[root@node<x>]# chown s4hadm:sapsys /usr/sap/S4H/D<Ins#>
```

3.4.5. Verifying that the SAP instances can be started and stopped on all cluster nodes

Stop the **(A)SCS** and **ERS** instances using **sapcontrol**, unmount the instance specific directories and then mount them on the other node:

```
/usr/sap/S4H/ASCS20/  
/usr/sap/S4H/ERS29/  
/usr/sap/S4H/D<Ins#>/
```

Verify that manual starting and stopping of all SAP instances using **sapcontrol** works on all HA cluster nodes and that the SAP instances are running correctly using the tools provided by SAP.

3.4.6. Verifying that the correct version of SAP Host Agent is installed on all HA cluster nodes

Run the following command on each cluster node to verify that SAP Host Agent has the same version and meets the minimum version requirement:

```
[root@node<x>]# /usr/sap/hostctrl/exe/saphostexec -version
```

Please check SAP Note [1031096-Installing Package SAPHOSTAGENT](#) in case SAP Host Agent needs to be updated.

3.4.7. Installing permanent SAP license keys

To ensure that the SAP instances continue to run after a failover, it might be necessary to install several SAP license keys based on the hardware key of each cluster node. Please see SAP Note [1178686 - Linux: Alternative method to generate a SAP hardware key](#) for more information.

3.4.8. Additional changes required when using systemd enabled SAP instances

If the SAP instances that will be managed by the cluster are [systemd enabled](#), additional configuration changes are required to ensure that **systemd** does not interfere with the management of the SAP instances by the HA cluster. Please check out section [2. Red Hat HA Solutions for SAP](#) in [The Systemd-Based SAP Startup Framework](#) for information.

CHAPTER 4. SETTING UP THE CLUSTER

4.1. PERFORMING THE BASIC CLUSTER INSTALLATION ON EACH NODE

Please refer to [Configuring and managing high availability clusters](#), to first set up a pacemaker cluster.

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 is available at [Cluster Platforms and Architectures](#).

The rest of this guide will assume that following things are working properly:

- Pacemaker cluster is configured according to documentation and has proper and working fencing (see [How to test fence devices and fencing configuration in a Red Hat High Availability cluster?](#), for information on procedures for verifying that fencing is working properly).
- Enqueue replication between the (A)SCS and ERS instances has been manually tested, as explained in [Setting up Enqueue Replication Server failover](#).
- All HA cluster nodes are subscribed to [RHEL for SAP Applications](#) or [RHEL for SAP Solutions](#), and the required repositories are enabled as explained in [RHEL for SAP Subscriptions and Repositories](#).

4.2. CONFIGURING GENERAL CLUSTER PROPERTIES

To avoid unnecessary failovers of the resources set the default values for the **resource-stickiness** and **migration-threshold** parameters by running the following commands on one cluster node:

```
[root@node1]# pcs resource defaults update resource-stickiness=1
[root@node1]# pcs resource defaults update migration-threshold=3
```



NOTE

The **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.

4.3. INSTALLING THE RESOURCE-AGENTS-SAP PACKAGE ON ALL CLUSTER NODES

The **SAPInstance** and **SAPDatabase** resource agents are provided via a separate **resource-agents-sap** package. Run the following command to install it on each HAcluster node:

```
[root@node<x>]# dnf install resource-agents-sap
```

4.4. CONFIGURING ACCESS TO SHARED FILE SYSTEMS

For the SAP instances to work, the following shared file systems must be available on all cluster nodes:

```
/sapmnt
/usr/sap/trans
```

The shared file systems can either be managed by the cluster or they can be statically mounted by adding them to the **/etc/fstab** on each cluster node.

4.4.1. Configuring shared file systems managed by the cluster

To mount the shares file systems from an external **NFS** server on all cluster nodes, cloned Filesystem cluster resources can be created as shown below:

```
[root@node1]# pcs resource create s4h_fs_sapmnt Filesystem device='<NFS_Server>:
<sapmnt_nfs_share>' directory='/sapmnt' fstype='nfs' clone interleave=true
[root@node1]# pcs resource create s4h_fs_sap_trans Filesystem device='<NFS_Server>:
<sap_trans_nfs_share>' directory='/usr/sap/trans' fstype='nfs' clone interleave=true
```

4.4.2. Configuring shared file systems managed outside of cluster

In case the shared file systems will not be managed by cluster, it must be ensured that they are available before the pacemaker service is started.

See [Chapter 13. Determining the order in which cluster resources are run](#) , for instructions on how to ensure that when the shared file systems are managed outside of the cluster, they are mounted before the cluster attempts to start any resources that require access to the shared file systems.

4.5. CONFIGURING THE (A)SCS RESOURCE GROUP

4.5.1. Creating resource for managing the virtual IP address of the (A)SCS instance

To allow application servers and other clients to connect to the **(A)SCS** instance on the HA cluster node where the instance is currently running, the virtual IP address (VIP) that has been assigned to the instance needs to be moved by the cluster when the **(A)SCS** instance is moving from one HA cluster node to another.

For this, a resource that manages the VIP needs to be created as part of the resource group that is used for managing the **(A)SCS** instance.

Please use the appropriate resource agent for managing the virtual IP address based on the platform on which the HA cluster is running.

On physical servers or VMs the resource can be created using the **IPaddr2** resource agent:

```
[root@node1]# pcs resource create s4h_vip_ascs20 IPaddr2 ip=192.168.200.101 --group
s4h_ASCS20_group
```

4.5.2. Creating resource for managing the (A)SCS instance directory

Since SAP requires that the instance directory be only available on the HA cluster node where the instance is supposed to be running it is necessary to set up HA cluster resources for managing the filesystems that are used for the instance directories.



NOTE

Even if the instance directories are stored on **NFS** it is still necessary to create the resource to allow the HA cluster to only mount the **NFS** export on the HA cluster node where the SAP instance should be running.

4.5.2.1. NFS

If the instance directory for the **(A)SCS** instance is located on **NFS**, the resource to have it managed as part of the resource group for managing the **(A)SCS** instance can be created with the following command:

```
[root@node1]# pcs resource create s4h_fs_ascscs20 Filesystem device='<NFS_Server>:
<s4h_ascscs20_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
```

4.5.2.2. HA-LVM

When using **HA-LVM** to manage the instance directory for the **(A)SCS** instance, it must be configured according to the guidelines in the article [What is a Highly Available LVM \(HA-LVM\) configuration and how do I implement it?](#).

First, an **LVM-activate** cluster resource must be added, followed by a Filesystem cluster resource:

```
[root@node1]# pcs resource create s4h_fs_ascscs20_lvm LVM-activate
volgrpname='<ascscs_volume_group>' vg_access_mode=system_id --group s4h_ASCS20_group

[root@node1]# pcs resource create s4h_fs_ascscs20 Filesystem
device='/dev/mapper/<ascscs_logical_volume>' directory=/usr/sap/S4H/ASCS20 fstype=ext4 --group
s4h_ASCS20_group
```

4.5.3. Creating resource for managing the (A)SCS instance

```
[root@node1]# pcs resource create s4h_ascscs20 SAPInstance
InstanceName="S4H_ASCS20_rhascs"
START_PROFILE=/sapmnt/S4H/profile/S4H_ASCS20_rhascs AUTOMATIC_RECOVER=false \
  meta resource-stickiness=5000 migration-threshold=1 \
  --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
```

resource-stickiness=5000 is used to balance out the failover constraint with the **ERS** resource so the resource stays on the node where it started and doesn't migrate around the cluster uncontrollably.

migration-threshold=1 ensures that the **(A)SCS** instance fails over to another node when an issue is detected instead of restarting on the same HA cluster node. For **ENSA2** setups this option is not required, since with **ENSA2** restarting the **(A)SCS** instance on the same HA cluster node is allowed.

When all resources for the resource group have been created, add a resource stickiness to the group to ensure that the **(A)SCS** instance will stay on a HA cluster node if possible:

```
[root@node1]# pcs resource meta s4h_ASCS20_group resource-stickiness=3000
```

4.6. CONFIGURING THE ERS RESOURCE GROUP

4.6.1. Creating resource for managing the virtual IP address of the ERS instance

Even though the **ERS** instance is not directly accessed by application servers, it still requires a virtual IP to allow SAP management tools to connect to the **ERS** instance on the HA cluster node where the instance is currently running. Therefore, the virtual IP address (VIP) that has been assigned to the instance needs to be moved by the cluster when the **(A)SCS** instance is moving from one HA cluster node to another.

For this, a resource that manages the VIP needs to be created as part of the resource group that is used for managing the **ERS** instance.

Please use the appropriate resource agent for managing the virtual IP address based on the platform on which the HA cluster is running.

On physical servers or VMs the resource can be created using the **IPaddr2** resource agent:

```
[root@node1]# pcs resource create s4h_vip_ers29 IPaddr2 ip=192.168.200.102 --group
s4h_ERS29_group
```

4.6.2. Creating resource for managing the ERS instance directory

Since SAP requires that the instance directory be only available on the HA cluster node where the instance is supposed to be running it is necessary to set up HA cluster resources for managing the filesystems that are used for the instance directories.



NOTE

Even if the instance directories are stored on **NFS** it is still necessary to create the resource to allow the HA cluster to only mount the **NFS** export on the HA cluster node where the SAP instance should be running.

4.6.2.1. NFS

If the instance directory for the **ERS** instance is located on **NFS**, the resource to have it managed as part of the resource group for managing the ERS instance can be created with the following command:

```
[root@node1]# 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
```

4.6.2.2. HA-LVM

When using **HA-LVM** to manage the instance directory for the **ERS** instance it must be configured according to the guidelines in the article [What is a Highly Available LVM \(HA-LVM\) configuration and how do I implement it?](#).

First, an **LVM-activate** cluster resource must be added, followed by a Filesystem cluster resource:

```
[root@node1]# pcs resource create s4h_fs_ers29_lvm LVM-activate
volgrpname='<ers_volume_group>' vg_access_mode=system_id --group s4h_ERS29_group

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

4.6.3. Creating resource for managing the ERS instance

Create the ERS instance cluster resource:

```
[root@node1]# pcs resource create s4h_ers29 SAPInstance InstanceName="S4H_ERS29_rhers"
START_PROFILE=/sapmnt/S4H/profile/S4H_ERS29_rhers AUTOMATIC_RECOVER=false
IS_ERS=true --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
```



NOTE

IS_ERS=true attribute is mandatory for **ENSA1** deployment. More information about **IS_ERS** can be found in [How does the IS_ERS attribute work on a SAP NetWeaver cluster with Standalone Enqueue Server \(ENSA1 and ENSA2\)?](#).

4.7. CREATING CONSTRAINTS

4.7.1. Creating colocation constraint for (A)SCS and ERS resource groups

Resource groups **s4h_ASCS20_group** and **s4h_ERS29_group** should try to avoid running on the same node. The order of the groups matters.

```
[root@node1]# pcs constraint colocation add s4h_ERS29_group with s4h_ASCS20_group -5000
```

4.7.2. Creating location constraint for (A)SCS resource (ENSA1 only)

When using **ENSA1**, it must be ensured that the **(A)SCS** instance moves to the node where the **ERS** instance is running when failover is happening.

```
[root@node1]# pcs constraint location s4h_ascs20 rule score=2000 runs_ers_S4H eq 1
```

4.7.3. Creating order constraint for (A)SCS and ERS resource groups

Stop the resource group for managing the **ERS** instance after the resource group for managing the **(A)SCS** instance has been started, if pacemaker decides to start the resource group for managing the **(A)SCS** instance at the same time as stopping the resource group for managing the **ERS** instance:

■

```
[root@node1]# pcs constraint order start s4h_ASCS20_group then stop s4h_ERS29_group
symmetrical=false kind=Optional
```



NOTE

Since **symmetrical=false** and **kind=Optional** are used, there can be situation where this constraint won't take effect. For more information, refer to [Determining the order in which cluster resources are run](#).

4.7.4. Creating order constraints for /sapmnt resource managed by cluster

If the shared filesystem **/sapmnt** is managed by the cluster, then the following constraints ensure that resource groups used for managing the **(A)SCS** and **ERS** instances are started only after the **/sapmnt** filesystem is available:

```
[root@node1]# pcs constraint order s4h_fs_sapmnt-clone then s4h_ASCS20_group
[root@node1]# pcs constraint order s4h_fs_sapmnt-clone then s4h_ERS29_group
```

4.8. CONFIGURING CLUSTER RESOURCE GROUP FOR MANAGING DATABASE INSTANCES (OPTIONAL)

When using the HA cluster for managing a SAP NetWeaver based SAP product that is still using a legacy database like Oracle, IBM DB2, SAP ASE or SAP MaxDB, it is possible to also have the database instance managed by the cluster.

This chapter shows an example of how to set up a resource group for managing a single database instance using the **SAPDatabase** resource agent and the virtual IP address and the file system required by it.

The example setup described in this chapter uses the **SAPSID RH1** instead of S4H, since the **SAPDatabase** resource agent cannot be used with S/4HANA setups.

4.8.1. Creating resource for managing the virtual IP address of the database instance

To create the resource for managing the virtual IP address used for accessing the database instance that will be part of the **rh1_SAPDatabase_group** use the following command:

```
[root]# pcs resource create rh1_vip_db IPAddr2 ip=192.168.200.115 --group
rh1_SAPDatabase_group
```

4.8.2. Creating resource for managing the directories used by the database instance

The directories used by the database instance can only be mounted on the HA cluster node where the database instance is supposed to run to avoid that the database instance can accidentally be started on another system at the same time which would cause data corruption.

Depending on the way the storage for managing the directories used by the database instance is set up, different methods for creating the resources for managing the database directories have to be used.



NOTE

Even if the instance directories are stored on **NFS** it is still necessary to create the resource to allow the HA cluster to only mount the **NFS** export on the HA cluster node where the database instance should be running.

4.8.2.1. NFS

If the directories used by the database instance are located on **NFS**, a resource to have them managed as part of the resource group for managing the database instance must be created for each directory with the following command:

```
[root@node1]# pcs resource create rh1_fs_db Filesystem device='<NFS_Server>:
<rh1_db_nfs_share>' directory=/sapdb/RH1 fstype=nfs force_unmount=safe --group
rh1_SAPDatabase_group \
  op start interval=0 timeout=60 \
  op stop interval=0 timeout=120 \
  op monitor interval=200 timeout=40
```

4.8.2.2. HA-LVM

When using **HA-LVM** to manage the directories used by the database instance it must be configured according to the guidelines in the article [What is a Highly Available LVM \(HA-LVM\) configuration and how do I implement it?](#).

First, an **LVM-activate** cluster resource must be added followed by a Filesystem cluster resource:

```
[root]# pcs resource create rh1_lvm_db LVM-activate volgrpname=vg_db
vg_access_mode=system_id --group rh1_SAPDatabase_group
[root]# pcs resource create rh1_fs_db Filesystem device=/dev/vg_db/lv_db directory=/sapdb/RH1
fstype=xfs --group rh1_SAPDatabase_group
```

If multiple file systems are used for the database directories, then a separate Filesystem cluster resource must be created for each one.

4.8.3. Configuring SAPDatabase cluster resource

After the resources for the virtual IP address and the filesystems required by the database instance have been added, the **SAPDatabase** cluster resource that will manage the database instance can be added to the resource group:

```
[root]# pcs resource create rh1_SAPDatabase SAPDatabase DBTYPE="ADA" SID="RH1"
STRICT_MONITORING="TRUE" AUTOMATIC_RECOVER="TRUE" --group
rh1_SAPDatabase_group
```

4.9. CONFIGURING PRIMARY/ADDITIONAL APPLICATION SERVER (PAS/AAS) RESOURCE GROUP (OPTIONAL)

This section describes how to configure a resource group for managing the Primary Application Server (PAS) instance and the associated VIP and filesystem for the instance directory, in case the PAS instance should also be managed by the HA cluster. The same configuration can also be used for Additional Application Server (AAS) instances that should be managed by the HA cluster.

4.9.1. Creating resource for managing the Virtual IP address (VIP) for the PAS/AAS instance

To allow other application servers and clients to PAS/AAS instances managed by the HA cluster, the virtual IP address (VIP) that has been assigned to the instance needs to be moved by the cluster when a PAS/AAS instance is moving from one HA cluster node to another.

For this, a resource that manages the VIP needs to be created as part of the resource group that is used for managing a PAS/AAS instance.

Please use the appropriate resource agent for managing the virtual IP address based on the platform on which the HA cluster is running.

On physical servers or VMs the resource can be created using the **IPaddr2** resource agent:

```
[root@node1]# pcs resource create s4h_vip_pas_d21 IPaddr2 ip=192.168.200.103 --group
s4h_PAS_D21_group
```

4.9.2. Creating resource for managing the filesystem for the PAS/AAS instance directory

Since SAP requires that the instance directory be only available on the HA cluster node where the instance is supposed to be running it is necessary to set up HA cluster resources for managing the filesystems that are used for the instance directories.



NOTE

Even if the instance directories are stored on **NFS** it is still necessary to create the resource to allow the HA cluster to only mount the **NFS** export on the HA cluster node where the SAP instance should be running.

4.9.2.1. NFS

If the instance directory for the PAS/AAS instance is located on **NFS**, the resource to have it managed as part of the resource group for managing the PAS/AAS instance can be created with the following command:

```
[root@node1]# pcs resource create s4h_fs_pas_d21 Filesystem device='<NFS_Server>:
<s4h_pas_d21_nfs_share>' directory=/usr/sap/S4H/D21 fstype=nfs force_unmount=safe --group
s4h_PAS21_D21_group \
  op start interval=0 timeout=60 \
  op stop interval=0 timeout=120 \
  op monitor interval=200 timeout=40
```

4.9.2.2. HA-LVM

When using **HA-LVM** to manage the instance directory for the PAS/AAS instance it must be configured according to the guidelines in the article [What is a Highly Available LVM \(HA-LVM\) configuration and how do I implement it?](#).

First, an LVM-activate cluster resource must be added followed by a Filesystem cluster resource:

```
[root@node1]# pcs resource create s4h_lvm_pas_d21 LVM-activate volgrpname=vg_d21
```

```
vg_access_mode=system_id --group s4h_PAS_D21_group
[root@node1]# pcs resource create s4h_fs_pas_d21 Filesystem device=/dev/vg_d21/lv_d21
directory=/usr/sap/S4H/D21 fstype=xfst --group s4h_PAS_D21_group
```

4.9.3. Configuring PAS/AAS SAPInstance cluster resource

To enable pacemakers to manage a PAS or AAS instance, the same **SAPInstance** resource agent as for **(A)SCS/ERS** instance can be used. The PAS/AAS instance is, compared to the **(A)SCS/ERS** instance setup, a simple instance and requires less attributes to be configured.

Check the command below for an example on how to create a PAS instance for the **D21** instance and place it at the end of the **s4h_PAS_D21_group** resource group.

```
[root@node1]# pcs resource create s4h_pas_d21 SAPInstance InstanceName="S4H_D21_s4h-pas"
DIR_PROFILE=/sapmnt/S4H/profile START_PROFILE=/sapmnt/S4H/profile/S4H_D21_s4h-pas --
group s4h_PAS_D21_group
```

4.9.4. Configuring constraints

4.9.4.1. Configuring order constraints for the PAS/AAS resource group(s)

The PAS/AAS instance(s) require the **(A)SCS** and database instance to be running before it can start properly. The following sections show how to set up the required constraints based on the different types of database instances that can be used by SAP NetWeaver / S/4HANA.

4.9.4.1.1. Deployments with s4h_SAPDatabase_group

For configurations that have one cluster resource group, it will start all resources needed by the database. For example, here the **SAPDatabase** resource agent is used to manage the database and is part of the database group **rh1_SAPDatabase_group**. Commands below will create constraints that will start the whole **rh1_PAS_D21_group** only once the **(A)SCS** instance was promoted and when the database group **rh1_SAPDatabase_group** is running.

```
[root@node1]# pcs constraint order rh1_SAPDatabase_group then rh1_PAS_D21_group
kind=Optional symmetrical=false
[root@node1]# pcs constraint order start rh1_ASCS20_group then rh1_PAS_D21_group
kind=Optional symmetrical=false
```

4.9.4.1.2. Deployments with SAP HANA with System Replication as database

When using SAP HANA database that is configured for system replication (SR) that is managed by cluster, the following constraints will ensure that the whole **s4h_PAS_D21_group** group will start only once the **(A)SCS** instance was promoted and when the SAP HANA **SAPHana_S4H_02-master** was promoted.

```
[root@node1]# pcs constraint order promote SAPHana_S4H_02-master then s4h_PAS_D21_group
Kind=Optional symmetrical=false
[root@node1]# pcs constraint order start s4h_ASCS20_group then s4h_PAS_D21_group
Kind=Optional symmetrical=false
```

4.9.4.2. Configuring colocation constraint for PAS and AAS SAPInstance cluster resources (Optional)

To ensure that PAS and AAS instances do not run on the same nodes whenever both nodes are running, you can add a negative colocation constraint with the command below:

```
[root@node1]# pcs constraint colocation add s4h_AAS_D22_group with s4h_PAS_D21_group
score=-1000
```

The score of **-1000** is to ensure that if only 1 node is available, the PAS/AAS instance will continue to run on the remaining 1 node. In such a situation, if you would like to keep the AAS instance down, then you can use the **score=-INFINITY** which will enforce this condition.

4.9.4.3. Creating order constraint for /sapmnt resource managed by cluster

If the shared filesystem **/sapmnt** is managed by the cluster, then the following constraint ensures that resource groups used for managing the PAS/AAS instance(s) are started only after the **/sapmnt** filesystem is available:

```
[root@node1]# pcs constraint order s4h_fs_sapmnt-clone then s4h_PAS_D21_group
```

4.10. STANDALONE ENQUEUE SERVER 2 (ENSA2) MULTI-NODE CLUSTER (OPTIONAL)

For SAP S/4HANA with ENSA2, more than two HA cluster nodes can be used for managing the **ASCS** and **ERS** instances. Please use the guidelines in the following section in case an additional cluster node should be added to allow more flexibility for the instances to failover in case there is an issue with the node they were running on.

4.10.1. OS Configuration

Create a node that's identical to the first two nodes, in terms of resources, subscriptions, OS configuration, etc.

In the example, the hostname of the node is node3. Make sure the `/etc/hosts` file on each cluster node contains the hostnames and IP addresses of all cluster nodes and also the virtual hostnames and virtual IP addresses of all SAP instances that are managed by the HA cluster.

Make sure to copy the SAP related entries in **/etc/services** from one of the first two nodes to the third node.

4.10.2. Creating users and groups

Create the users and groups required by the SAP instances identical to the ones used on the other nodes. For example:

```
Groups in /etc/group:
sapsys:x:1010:
sapinst:x:1011:root,s4hadm
```

```
Users in /etc/passwd:
s4hadm:x:1020:1010:SAP System Administrator:/home/s4hadm:/bin/csh
sapadm:x:1001:1010:SAP System Administrator:/home/sapadm:/bin/false
```

4.10.3. Creating local directories and mount points for the shared file systems

Create all mount points required for all instances that should be able to run on the additional HA cluster node:

```
/sapmnt
/usr/sap/
/usr/sap/SYS/
/usr/sap/trans/
/usr/sap/S4H/
/usr/sap/S4H/ASCS20/
/usr/sap/S4H/ERS29/
/usr/sap/S4H/D<Ins#>/
```

Make sure to set the user and group ownership for all directories to the same user and group as on the other cluster nodes and copy the contents of the local directories (e. g., **/usr/sap/SYS**) from one of the other cluster nodes.

If **/sapmnt** and **/usr/sap/trans** are statically mounted on the existing HA cluster nodes via **/etc/fstab** then these file systems must also be added to the **/etc/fstab** on the additional HA cluster node and the file systems must be mounted afterwards.

If **/sapmnt** and **/usr/sap/trans** are managed by the cluster then the cluster configuration must be updated so that the file systems will also be mounted on the additional HA cluster node.

4.10.4. Installing the RHEL HA Add-On and the resource agents for managing SAP instances

For the node to be part of the cluster and to be able to manage the SAP instances, install the required packages:

```
[root@node3]# dnf install pcs pacemaker resource-agents-sap
```

4.10.5. Adding the node to the cluster

On one node of the existing cluster add the third node:

```
[root@node1]# pcs cluster auth node3
Username: hacluster
Password:

[root@node1]# pcs cluster node add node3
```

4.10.6. Updating fencing/STONITH configuration to include the 3rd node

Depending on the STONITH setup, you may need to update the STONITH resource to include the 3rd HA cluster node.

Before moving any resources to the new HA cluster node, please use the following command to verify that it is possible to fence the HA cluster new node from one of the existing HA cluster nodes:

```
[root@node1]# pcs stonith fence node3
```

4.10.7. Updating ERS resource configuration

To ensure that the **ERS** instance stays on the node where it started and doesn't migrate around the cluster uncontrollably, set **resource-stickiness** for the resource:

```
[root@node1]# pcs resource meta s4h_ers29 \ resource-stickiness=3000
```

4.11. ENABLING THE SAP HA INTERFACE TO ALLOW SAP INSTANCES CONTROLLED BY THE CLUSTER TO BE MANAGED BY SAP MANAGEMENT TOOLS (OPTIONAL)

To allow SAP admins to manage the SAP application server instances that are controlled by the HA cluster setup described in this documentation using tools like [SAP Landscape Management \(LaMa\)](#), the [SAP HA interface](#) must be enabled for each SAP application server instance managed by the HA cluster to ensure that the HA cluster is aware of any action performed by the SAP management tools that will affect the cluster resources used to manage the SAP instances (for example, the HA cluster needs to be notified if a SAP app server instance it manages is being started or stopped via SAP LaMa).

Please check [How to enable the SAP HA Interface for SAP ABAP application server instances managed by the RHEL HA Add-On?](#), for instructions on how to configure the SAP HA interface.

4.12. ENABLING CLUSTER TO AUTO-START AT BOOT (OPTIONAL)

By default the HA cluster is not enabled to auto-start at the boot of the OS, and it needs to be manually started after a cluster node is fenced and rebooted.

The automatic start of all cluster components on all cluster nodes can be enabled with the following command:

```
[root@node1]# 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, the filesystem needs to be repaired first before it can be used again. Having the cluster auto-start, but then fail because the filesystem doesn't work, can cause even more trouble.

CHAPTER 5. TESTING THE CLUSTER CONFIGURATION

Before the HA cluster setup is put in production, it is recommended to perform the following tests to ensure that the HA cluster setup works as expected.

These tests should also be repeated later on as part of regular HA/DR drills to ensure that the cluster still works as expected and that admins stay familiar with the procedures required to bring the setup back to a healthy state in case an issue occurs during normal operation, or if manual maintenance of the setup is required.

5.1. MANUALLY MOVING **ascs** INSTANCE USING **pcs** COMMAND

To verify that the pacemaker cluster is able to move the instances to the other HA cluster node on demand.

- Test Preconditions
 - Both cluster nodes are up, with the resource groups for the **ASCS** and **ERS** running on different HA cluster nodes:


```
* Resource Group: S4H_ASCS20_group:
  * S4H_lvm_ascs20 (ocf:heartbeat:LVM-activate): Started node1
  * S4H_fs_ascs20 (ocf:heartbeat:Filesystem): Started node1
  * S4H_vip_ascs20 (ocf:heartbeat:IPAddr2): Started node1
  * S4H_ascs20 (ocf:heartbeat:SAPInstance): Started node1
* Resource Group: S4H_ERS29_group:
  * S4H_lvm_ers29 (ocf:heartbeat:LVM-activate): Started node2
  * S4H_fs_ers29 (ocf:heartbeat:Filesystem): Started node2
  * S4H_vip_ers29 (ocf:heartbeat:IPAddr2): Started node2
  * S4H_ers29 (ocf:heartbeat:SAPInstance): Started node2
```
 - All failures for the resources and resource groups have been cleared and the failcounts have been reset.
- Test Procedure
 - Run the following command from any node to initiate the move of the **ASCS** instance to the other HA cluster node:


```
[root@node1]# pcs resource move S4H_ascs20
```
- Monitoring
 - Run the following command in a separate terminal during the test:


```
[root@node2]# watch -n 1 pcs status
```
- Expected behavior
 - The **ASCS** resource group is moved to the other node.
 - The **ERS** resource group stops after that and moves to the node where the **ASCS** resource group was running before.
- Test Result

- **ASCS** resource group moves to other node, in this scenario node node2 and **ERS** resource group moves to node node1:

```
* Resource Group: S4H_ASCS20_group:
* S4H_lvm_asc20 (ocf:heartbeat:LVM-activate): Started node2
* S4H_fs_asc20 (ocf:heartbeat:Filesystem): Started node2
* S4H_vip_asc20 (ocf:heartbeat:IPaddr2): Started node2
* S4H_asc20 (ocf:heartbeat:SAPInstance): Started node2
* Resource Group: S4H_ERS29_group:
* S4H_lvm_ers29 (ocf:heartbeat:LVM-activate): Started node1
* S4H_fs_ers29 (ocf:heartbeat:Filesystem): Started node1
* S4H_vip_ers29 (ocf:heartbeat:IPaddr2): Started node1
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node1
```

- Recovery Procedure:
 - Remove the location constraints, if any:

```
[root@node1]# pcs resource clear S4H_asc20
```

5.2. MANUALLY MOVING OF THE asc20 INSTANCE USING sapcontrol (WITH SAP HA INTERFACE ENABLED)

To verify that the **sapcontrol** command is able to move the instances to the other HA cluster node when the [SAP HA interface](#) is enabled for the instance.

- Test Preconditions
 - The [SAP HA interface](#) is enabled for the SAP instance.
 - Both cluster nodes are up with the resource groups for the **ASCS** and **ERS** running.

```
[root@node2: ~]# pcs status | egrep -e "S4H_asc20|S4H_ers29"
* S4H_asc20 (ocf:heartbeat:SAPInstance): Started node2
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node1
```

- All failures for the resources and resource groups have been cleared and the failcounts have been reset.
- Test Procedure
 - As the **<sid>adm** user, run the **HAFailoverToNode** function of **sapcontrol** to move the **ASCS** instance to the other node.
- Monitoring
 - Run the following command in a separate terminal during the test:

```
[root@node2]# watch -n 1 pcs status
```

- Expected behavior
 - **ASCS** instances should move to the other HA cluster node, creating a temporary location constraint for the move to complete.

- Test

```
[root@node2]# su - s4hadm
node2:s4hadm 52> sapcontrol -nr 20 -function HAFailoverToNode ""

06.12.2023 12:57:04
HAFailoverToNode
OK
```

- Test result

- **ASCS** and **ERS** both move to the other node:

```
[root@node2]# pcs status | egrep -e "S4H_ascs20|S4H_ers29"
* S4H_ascs20 (ocf:heartbeat:SAPInstance): Started node1
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node2
```

- Constraints are created as shown below:

```
[root@node1]# pcs constraint
Location Constraints:
Resource: S4H_ASCS20_group
Constraint: cli-ban-S4H_ASCS20_group-on-node2
Rule: boolean-op=and score=-INFINITY
Expression: #uname eq string node1
Expression: date lt xxxx-xx-xx xx:xx:xx +xx:xx
```

- Recovery Procedure

- The constraint shown above is cleared automatically when the **date lt** mentioned in the Expression is reached.
- Alternatively, the constraint can be removed with the following command:

```
[root@node1]# pcs resource clear S4H_ascs20
```

5.3. TESTING FAILURE OF THE **asc**s INSTANCE

To verify that the pacemaker cluster takes necessary action when the enqueue server of the **ASCS** instance or the whole **ASCS** instance fails.

- Test Preconditions

- Both cluster nodes are up with the resource groups for the **ASCS** and **ERS** running:

```
[root@node2]# pcs status | egrep -e "S4H_ascs20|S4H_ers29"
* S4H_ascs20 (ocf:heartbeat:SAPInstance): Started node1
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node2
```

- All failures for the resources and resource groups have been cleared and the failcounts have been reset.

- Test Procedure

- Identify the **PID** of the enqueue server on the node where **ASCS** is running.
- Send a **SIGKILL** signal to the identified process.
- Monitoring
 - Run the following command in a separate terminal during the test:


```
[root@node2]# watch -n 1 pcs status
```
- Expected behavior
 - Enqueue server process gets killed.
 - The pacemaker cluster takes the required action as per configuration, in this case moving the **ASCS** to the other node.
- Test
 - Switch to the **<sid>adm user** on the node where **ASCS** is running:


```
[root@node1]# su - s4hadm
```
 - Identify the PID of en.sap(NetWeaver) enq.sap(S/4HANA):


```
node1:s4hadm 51> pgrep -af "(en|enq).sap"
31464 enq.sapS4H_ASCS20 pf=/usr/sap/S4H/SYS/profile/S4H_ASCS20_s4ascs
```
 - Kill the identified process:


```
node1:s4hadm 52> kill -9 31464
```
 - Notice the cluster **Failed Resource Actions**:


```
[root@node2]# pcs status | grep "Failed Resource Actions" -A1
Failed Resource Actions:
* S4H_ascs20 2m-interval monitor on node1 returned 'not running' at Wed Dec 6
15:37:24 2023
```
 - **ASCS** and **ERS** move to the other node:


```
[root@node2]# pcs status | egrep -e "S4H_ascs20|S4H_ers29"
* S4H_ascs20 (ocf:heartbeat:SAPInstance): Started node2
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node1
* S4H_ascs20 2m-interval monitor on node1 returned 'not running' at Wed Dec 6
15:37:24 2023
```
- Recovery Procedure
 - Clear the failed action:


```
[root@node2]# pcs resource cleanup S4H_ascs20
...
Waiting for 1 reply from the controller
... got reply (done)
```

5.4. TESTING FAILURE OF THE ERS INSTANCE

To verify that the pacemaker cluster takes necessary action when the enqueue replication server (**ERS**) of the **ASCS** instance fails.

- Test Preconditions

- Both cluster nodes are up with the resource groups for the **ASCS** and **ERS** running:

```
[root@node1]# pcs status | egrep -e "S4H_ascs20|S4H_ers29"
* S4H_ascs20 (ocf:heartbeat:SAPInstance): Started node2
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node1
```

- All failures for the resources and resource groups have been cleared and the failcounts have been reset.

- Test Procedure

- Identify the PID of the enqueue replication server process on the node where the **ERS** instance is running.
- Send a SIGKILL signal to the identified process.

- Monitoring

- Run the following command in a separate terminal during the test:

```
[root@node2]# watch -n 1 pcs status
```

- Expected behavior

- Enqueue Replication server process gets killed.
- Pacemaker cluster takes the required action as per configuration, in this case, restarting the **ERS** instance on the same node.

- Test

- Switch to the **<sid>adm** user:

```
[root@node1]# su - s4hadm
```

- Identify the PID of **enqr.sap**:

```
node1:s4hadm 56> pgrep -af enqr.sap
532273 enqr.sapS4H_ERS29 pf=/usr/sap/S4H/SYS/profile/S4H_ERS29_s4ers
```

- Kill the identified process:

```
node1:s4hadm 58> kill -9 532273
```

- Notice the cluster "Failed Resource Actions":

```
[root@node1]# pcs status | grep "Failed Resource Actions" -A1
```

```
Failed Resource Actions:
```

```
* S4H_ers29 2m-interval monitor on node1 returned 'not running' at Thu Dec 7
13:15:02 2023
```

- **ERS** restarts on the same node without disturbing the **ASCS** already running on the other node:

```
[root@node1]# pcs status | egrep -e "S4H_ascs20|S4H_ers29"
* S4H_ascs20 (ocf:heartbeat:SAPInstance): Started node2
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node1
* S4H_ers29 2m-interval monitor on node1 returned 'not running' at Thu Dec 7
13:15:02 2023
```

- Recovery Procedure

- Clear the failed action:

```
[root@node1]# pcs resource cleanup S4H_ers29
...
Waiting for 1 reply from the controller
... got reply (done)
```

5.5. FAILOVER OF ASCS INSTANCE DUE TO NODE CRASH

To verify that the **ASCS** instance moves correctly in case of a node crash.

- Test Preconditions

- Both cluster nodes are up with the resource groups for the **ASCS** and **ERS** running:

```
[root@node1]# pcs status | egrep -e "S4H_ascs20|S4H_ers29"
* S4H_ascs20 (ocf:heartbeat:SAPInstance): Started node2
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node1
```

- All failures for the resources and resource groups have been cleared and the failcounts have been reset.

- Test Procedure

- Crash the node where **ASCS** is running.

- Monitoring

- Run the following command in a separate terminal on the other node during the test:

```
[root@node1]# watch -n 1 pcs status
```

- Expected behavior

- Node where **ASCS** is running gets crashed and shuts down or restarts as per configuration.
- Meanwhile **ASCS** moves to the other node.
- **ERS** starts on the previously crashed node, after it comes back online.

- Test
 - Run the following command as the root user on the node where **ASCS** is running:

```
[root@node2]# echo c > /proc/sysrq-trigger
```

- **ASCS** moves to the other node:

```
[root@node1]# pcs status | egrep -e "S4H_asc20|S4H_ers29"
* S4H_asc20 (ocf:heartbeat:SAPInstance): Started node1
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node1
```

- **ERS** stops and moves to the previously crashed node once it comes back online:

```
[root@node1]# pcs status | egrep -e "S4H_asc20|S4H_ers29"
* S4H_asc20 (ocf:heartbeat:SAPInstance): Started node1
* S4H_ers29 (ocf:heartbeat:SAPInstance): Stopped
```

```
[root@node1]# pcs status | egrep -e "S4H_asc20|S4H_ers29"
* S4H_asc20 (ocf:heartbeat:SAPInstance): Started node1
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node2
```

- Recovery Procedure
 - Clean up failed actions, if any:

```
[root@node1]# pcs resource cleanup
```

5.6. FAILURE OF ERS INSTANCE DUE TO NODE CRASH

To verify that the **ERS** instance restarts on the same node.

- Test Preconditions
 - Both cluster nodes are up with the resource groups for the **ASCS** and **ERS** running:

```
[root@node1]# pcs status | egrep -e "S4H_asc20|S4H_ers29"
* S4H_asc20 (ocf:heartbeat:SAPInstance): Started node1
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node2
```

- All failures for the resources and resource groups have been cleared and the failcounts have been reset.

- Test Procedure

- Crash the node where **ERS** is running.

- Monitoring

- Run the following command in a separate terminal on the other node during the test:

```
[root@nod1]# watch -n 1 pcs status
```


- Expected behavior
 - Node where **ERS** is running gets crashed and shuts down or restarts as per configuration.
 - Meanwhile **ASCS** continues to run to the other node. **ERS** restarts on the crashed node, after it comes back online.
- Test
 - Run the following command as the root user on the node where **ERS** is running:


```
[root@node2]# echo c > /proc/sysrq-trigger
```
 - **ERS** restarts on the crashed node, after it comes back online, without disturbing the **ASCS** instance throughout the test:


```
[root@node1]# pcs status | egrep -e "S4H_ascs20|S4H_ers29"
* S4H_ascs20 (ocf:heartbeat:SAPInstance): Started node1
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node2
```
- Recovery Procedure
 - Clean up failed actions if any:


```
[root@node2]# pcs resource cleanup
```

5.7. FAILURE OF ASCS INSTANCE DUE TO NODE CRASH (ENSA2)

In case of 3 node ENSA 2 cluster environment, the third node is considered during failover events of any instance.

- Test Preconditions
 - A 3 node SAP S/4HANA cluster with the resource groups for the **ASCS** and **ERS** running.
 - The 3rd node has access to all the file systems and can provision the required instance specific IP addresses the same way as the first 2 nodes.
 - In the example setup, the underlying shared **NFS** filesystems are as follows:

```
Node List:
* Online: [ node1 node2 node3 ]

Active Resources:
* s4r9g2_fence (stonith:fence_rhevm): Started node1
* Clone Set: s4h_fs_sapmnt-clone [fs_sapmnt]:
* Started: [ node1 node2 node3 ]
* Clone Set: s4h_fs_sap_trans-clone [fs_sap_trans]:
* Started: [ node1 node2 node3 ]
* Clone Set: s4h_fs_sap_SYS-clone [fs_sap_SYS]:
* Started: [ node1 node2 node3 ]
* Resource Group: S4H_ASCS20_group:
* S4H_lvm_ascs20 (ocf:heartbeat:LVM-activate): Started node1
* S4H_fs_ascs20 (ocf:heartbeat:Filesystem): Started node1
* S4H_vip_ascs20 (ocf:heartbeat:IPAddr2): Started node1
```

```
* S4H_asc20 (ocf:heartbeat:SAPInstance): Started node1
* Resource Group: S4H_ERS29_group:
* S4H_lvm_ers29 (ocf:heartbeat:LVM-activate): Started node2
* S4H_fs_ers29 (ocf:heartbeat:Filesystem): Started node2
* S4H_vip_ers29 (ocf:heartbeat:IPaddr2): Started node2
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node2
```

- All failures for the resources and resource groups have been cleared and the failcounts have been reset.
- Test Procedure
 - Crash the node where **ASCS** is running.
- Monitoring
 - Run the following command in a separate terminal on one of the nodes where the **ASCS** group is currently not running during the test:

```
[root@node2]# watch -n 1 pcs status
```

- Expected behavior
 - **ASCS** moves to the 3rd node.
 - **ERS** continues to run on the same node where it is already running.
- Test
 - Crash the node where the **ASCS** group is currently running:

```
[root@node1]# echo c > /proc/sysrq-trigger
```

- **ASCS** moves to the 3rd node without disturbing the already running **ERS** instance on 2nd node:

```
[root@node2]# pcs status | egrep -e "S4H_asc20|S4H_ers29"
* S4H_asc20 (ocf:heartbeat:SAPInstance): Started node3
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node2
```

- Recovery Procedure
 - Clean up failed actions if any:

```
[root@node2]# pcs resource cleanup
```

CHAPTER 6. MAINTENANCE PROCEDURES

6.1. UPDATE RHEL AND THE RHEL HA ADD-ON

Please refer to [Recommendations: Applying package updates in a RHEL High Availability cluster](#) , for more information.



NOTE

For two-node cluster setups it is not necessary to manually move the resources to the other HA cluster node before placing the HA cluster node in standby mode (putting the HA cluster node in standby mode will take care of moving or stopping the resources running on the HA cluster node based on the HA cluster configuration).

Also, to minimize downtime of the SAP system it is recommended to first update the HA cluster node running the “less critical” resources, like the **ERS** instance. When the HA cluster node has been updated and it has been verified that the resources that were running on the node before the update was started are running again, the other HA cluster node running the “critical” resources, like the **(A)SCS** instance, can be updated as well.

CHAPTER 7. REFERENCES

7.1. RED HAT

- [Configuring and managing high availability clusters](#)
- [Support Policies for RHEL High Availability Clusters](#)
- [Support Policies for RHEL High Availability Clusters - Fencing/STONITH](#)
- [Support Policies for RHEL High Availability Clusters - Management of SAP S/4HANA](#)
- [Support Policies for RHEL High Availability Clusters - Management of SAP NetWeaver in a Cluster](#)
- [Red Hat HA Solutions for SAP HANA, S/4HANA and NetWeaver based SAP Applications](#)
- [How to enable the SAP HA Interface for SAP ABAP application server instances managed by the RHEL HA Add-On?](#)
- [How to manage standalone SAP Web Dispatcher instances using the RHEL HA Add-On](#)
- [The Systemd-Based SAP Startup Framework](#)

7.2. SAP

- [SAP Note 1552925 - Linux: High Availability Cluster Solutions](#)
- [SAP Note 1693245 - SAP HA Script Connector Library](#)
- [SAP Note 1908655 - Support details for Red Hat Enterprise Linux HA Add-On](#)
- [SAP Note 2630416 - Support for Standalone Enqueue Server 2](#)
- [SAP Note 2641322 - Installation of ENSA2 and update from ENSA1 to ENSA2 when using the Red Hat HA solutions for SAP](#)
- [SAP Note 2772999 - Red Hat Enterprise Linux 8.x: Installation and Configuration](#)
- [Standalone Enqueue Server | SAP Help Portal](#)
- [Setting up Enqueue Replication Server Fail over | SAP Blogs](#)
- [High Availability with the Standalone Enqueue Server](#)
- [Evolution of ENSA2 and ERS2... | SAP Blogs](#)