Configuring Cost-Optimized SAP S/4HANA ASCS/ERS/HDB with Standalone Enqueue Server 2 (ENSA 2) in a Pacemaker Cluster

SAP Alliance Team

Date: Apr 2022
3.4.1. ASCS profile modification
3.4.2. ERS profile modification
3.4.3. Update the /usr/sap/sap_services file
3.4.4. Create mount points for ASCS and ERS on the failover node
3.4.5. Manually Testing Instances on Other Node
3.4.6. Check SAP HostAgent on all nodes
3.4.7. Install permanent SAP license keys

4. Install Pacemaker
   4.1. Install Pacemaker rpm's
   4.2. Create a Cluster
      4.2.1. Define General Cluster Properties
   4.3. Configure STONITH
   4.4. Install resource-agents-sap on all cluster nodes
   4.5. Configure cluster resources for shared filesystems
      4.5.1. Configure shared filesystems managed by the cluster
   4.6. Configure ASCS resource group
      4.6.1. Create resource for virtual IP address
      4.6.2. Create resource for ASCS filesystem.
      4.6.3. Create resource for ASCS instance
   4.7. Configure ERS resource group
      4.7.1. Create resource for virtual IP address
      4.7.2. Create resource for ERS filesystem
      4.7.3. Create resource for ERS instance
   4.8. Create constraints
      4.8.1. Create colocation constraint for ASCS and ERS resource groups
      4.8.2. Create order constraint for ASCS and ERS resource groups
4.8.3. Create order constraint for /sapmnt resource managed by cluster

5. Test the cluster configuration
   5.1. Check the constraints
   5.2. Failover ASCS due to node crash
   5.3. ERS moves to the previously failed node

6. Enable cluster to auto-start after reboot

7. Test failover
   7.1. Failover ASCS due to node crash
   7.2.2. ERS remains on current node
   7.2.3. Test ERS crash

8. Optional - Enable SAP HA interface for Management of Cluster-controlled ASCS/ERS instances using SAP Management Tools
1. Overview

1.1. Introduction

Cost-Optimized SAP S/4HANA systems play an important role in business processes, 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.

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 Consulting 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 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 a SAP HANA Scale-Up instance, with fully automated failover using SAP HANA System Replication, where the master and the slave SAP HANA resources will run on each node as per set constraints. This article does NOT cover preparation of a 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 the above combined is considered as a Cost-Optimized SAP S/4HANA with an Automated SAP HANA Scale-Up System Replication Environment.

1.4. Support Policies


This solution is supported subject to fulfilling the above policies.

2. Requirements
2.1 Subscription

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

Follow this kbase article to enable the repos on both nodes which are required for this environment.

2.2. Pacemaker Resource Agents

- RHEL for SAP Solutions 8.x.

2.2.1 SAPInstance

SAPInstance pacemaker 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. SAP S/4HANA High-Availability Architecture

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

- SAP S/4HANA ASCS/ERS cluster resources
- Database: SAP HANA with System Replication
  - Configure Automated HANA System Replication in pacemaker

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 using the RHEL HA add-on.

2.4. 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.

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.

2.5. Storage requirements

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

2.5.1. 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.

2.5.2. 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:

- use of the external NFS server (NFS server cannot run on any of nodes inside the cluster in which the shares would be mounted, more details about this limitation can be found in 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.

3. Install SAP S/4HANA

3.1. Configuration options used in this document

Below are 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:

<table>
<thead>
<tr>
<th>Raw 1st node hostname: s4node1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2nd node hostname: s4node2</td>
</tr>
<tr>
<td>SID: S4H</td>
</tr>
<tr>
<td>ASCS Instance number: 20</td>
</tr>
<tr>
<td>ASCS virtual hostname: s4ascs</td>
</tr>
<tr>
<td>ERS Instance number: 29</td>
</tr>
</tbody>
</table>
ERS virtual hostname: s4ers

HANA database:

<table>
<thead>
<tr>
<th>Raw SID: S4D</th>
</tr>
</thead>
<tbody>
<tr>
<td>HANA Instance number: 00</td>
</tr>
<tr>
<td>HANA virtual hostname: s4db</td>
</tr>
</tbody>
</table>

3.2. Prepare hosts

Before starting installation ensure that:

- Install RHEL 8 for SAP Solutions (latest certified for SAP HANA is recommended)
- Register 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
- Shared storage and filesystems are present at correct mount points
- Virtual IP addresses used by instances are present and reachable
- Hostnames that will be used by instances can be resolved to IP addresses and back
- Installation medias are available
- System is configured according to the recommendation for running SAP S/4HANA
  - Red Hat Enterprise Linux 8.x: Installation and Configuration - SAP OSS Note 2772999

3.3. Install S/4HANA

Using SAP’s Software Provisioning Manager (SWPM) 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:

```
Raw
/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:

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

Virtual IP for s4ers should be enabled on s4node2.

Run the installer:

Raw
[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.

Raw
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:

```
[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:

```
[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 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

Raw
[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:

Raw
[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:

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

To upgrade/install SAP HostAgent, 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.

4. Install Pacemaker

Follow Pacemaker documentation:

Configuring and Managing High Availability Clusters on RHEL 8
How can I configure power fencing for the IBM POWER platform using an HMC in a RHEL High Availability cluster?
Below is a sample procedure to install a Pacemaker cluster. It’s recommended to work with a Red Hat consultant to install and configure Pacemaker in your environment.

4.1. Install Pacemaker rpm’s

Raw
# yum -y install pcs pacemaker
# passwd hacluster
[provide a password]
# systemctl enable –now pcsd.service

4.2. Create a Cluster

Create a cluster named s4ha, consisting of s4node1 and s4node2, and start the cluster. Please note that at this point, the cluster is not yet configured to auto-start after reboot.

Raw
# pcs cluster auth s4node1 s4node2
# pcs cluster setup --name s4ha s4node1 s4node2
# pcs cluster start --all

4.2.1. Define General Cluster Properties

Set the resource stickiness:

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

4.3. Configure STONITH

The fencing mechanism STONITH depends on the underlying platform. Please check the corresponding document to configure the STONITH. Support Policies for RHEL High Availability Clusters - General Requirements for Fencing/STONITH.

After configuring the STONITH, on s4node1 test by fencing s4node2 and vice-versa:

```
[root@s4node1]# pcs stonith fence s4node2
s4node2 should be properly fenced. After fencing, start cluster on s4node2 using the following command because the cluster has not yet been enabled to auto-start. Auto-start will be enabled after initial testing.
```

```
[root@s4node2 ~]# pcs cluster start
```

4.4. Install resource-agents-sap on all cluster nodes

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

4.5. Configure cluster resources for shared filesystems
Configure shared filesystem to provide following mount points on all the cluster nodes.

Raw
/sapmnt
/usr/sap/trans
/usr/sap/S4H/SYS

4.5.1. 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.

Raw
[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.

Raw
[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]
4.6. Configure ASCS resource group

4.6.1. Create resource for virtual IP address

Raw

# pcs resource create s4h_vip_asc20 IPaddr2 ip=192.168.200.201
--group s4h_ASCS20_group

4.6.2. Create resource for ASCS filesystem.

Below is the example of creating resource for NFS filesystem

Raw

# pcs resource create s4h_fs_asc20 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

Raw

# pcs resource create s4h_fs_asc20_lvm LVM
  volgrpname='<ascs_volume_group>' exclusive=true
  --group s4h_ASCS20_group

# pcs resource create s4h_fs_asc20 Filesystem
  device='/dev/mapper/<ascs_logical_volume>'
  directory=/usr/sap/S4H/ASCS20 fstype=ext4
  --group s4h_ASCS20_group
4.6.3. Create resource for ASCS instance

Raw

```bash
# pcs resource create s4h_asc20 SAPInstance \
InstanceName="S4H_ASCS20_s4ascs" \
START_PROFILE=/sapmnt/S4H/profile/S4H_ASCS20_s4ascs \
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:

Raw

```
# pcs resource meta s4h_ASCS20_group resource-stickiness=3000
```

4.7. Configure ERS resource group

4.7.1. Create resource for virtual IP address

Raw

```
# pcs resource create s4h_vip_ers29 IPaddr2 ip=192.168.200.202 \
--group s4h_ERS29_group
```

4.7.2. Create resource for ERS filesystem

```
Below is the example of creating resource for NFS filesystem

Raw

```bash
# 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

Raw

```bash
# pcs resource create s4h_fs_ers29_lvm LVM
volgrpname='<ers_volume_group>' exclusive=true --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
```

4.7.3. 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)?.

Raw

```bash
# 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
```
4.8. Create constraints

4.8.1. 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.

Raw

```
# pcs constraint colocation add s4h_ERS29_group with s4h_ASCS20_group
-5000
```

4.8.2. Create order constraint for ASCS and ERS resource groups

Prefer to start s4h_ASCS20_group before the s4h_ERS29_group

Raw

```
# pcs constraint order start s4h_ASCS20_group then start
s4h_ERS29_group symmetrical=false kind=Optional
# pcs constraint order start s4h_ASCS20_group then stop
s4h_ERS29_group symmetrical=false kind=Optional
```

4.8.3. 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.

Raw

```
# pcs constraint order s4h_fs_sapmnt-clone then s4h_ASCS20_group
# pcs constraint order s4h_fs_sapmnt-clone then s4h_ERS29_group
```

5. Test the cluster configuration
5.1. Check the constraints

Raw

# 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 running on s4node2.

Raw

# pcs status
...
Resource Group: s4h_ASCS20_group
s4h_fs_asc20 (ocf::heartbeat:Filesystem): Started s4node1
s4h_vip_asc20 (ocf::heartbeat:IPaddr2): Started s4node1
s4h_asc20 (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
...
```

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:

```
# 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 but then fail because the filesystem doesn’t work 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.

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_asc20 (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
...

7.2.2. ERS remains on current node

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

7.2.3. 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.

7.2.4. Your cost-optimized SAP S/4HANA Cluster environment can look like the one below:

[root@s4hana1 ~]# pcs status
Cluster name: SAP-S4-HANA

....
Node List:
  * Online: [ s4hana1 s4hana2 ]

....
Full List of Resources:
  * s4-fence (stonith:fence_rhevm): Started s4hana1
  * Clone Set: fs_sapmnt-clone [fs_sapmnt]:
    * Started: [ s4hana1 s4hana2 ]
  * Clone Set: fs_sap_trans-clone [fs_sap_trans]:
    * Started: [ s4hana1 s4hana2 ]
  * Clone Set: fs_sap_SYS-clone [fs_sap_SYS]:
    * Started: [ s4hana1 s4hana2 ]
  * Resource Group: s4h_ASCS20_group:
    * s4h_lvm_asc20 (ocf::heartbeat:LVM-activate): Started s4hana1
    * s4h_fs_asc20 (ocf::heartbeat:Filesystem): Started s4hana1
    * s4h_asc20 (ocf::heartbeat:SAPInstance): Started s4hana1
    * s4h_vip_asc20 (ocf::heartbeat:IPaddr2): Started s4hana1
  * Resource Group: s4h_ERS29_group:
    * s4h_lvm_ers29 (ocf::heartbeat:LVM-activate): Started s4hana2
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 configure SAP HAlib for SAPInstance resources?