



# SAP NetWeaver High Availability Cluster 7.40 - Setup Guide

---

Fabian Herschel Bernd Schubert

Publication Date: 2018/06/21

## Contents

- 1 About this Guide 2
- 2 Scope of this Document 4
- 3 Overview 5
- 4 SAP Installation 16
- 5 Implement the Cluster 23
- 6 Administration 29
- 7 Appendix 33

Revision 1.1 from 2018-06-21

SUSE LLC

10 Canal Park Drive

Suite 200

Cambridge MA 02142

USA

<http://www.suse.com/documentation> ↗

Copyright © 2018 SUSE LLC and contributors. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or (at your option) version 1.3; with the Invariant Section being this copyright notice and license. A copy of the license version 1.2 is included in the section entitled [GNU Free Documentation License](#).

For SUSE trademarks, see Trademark and Service Mark list <http://www.suse.com/company/legal/> ↗ Linux\* is a registered trademark of Linus Torvalds. All other third party trademarks are the property of their respective owners. A trademark symbol (®, ™ etc.) denotes a SUSE trademark; an asterisk (\*) denotes a third party trademark.

All information found in this book has been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. Neither SUSE LLC, the authors, nor the translators shall be held liable for possible errors or the consequences thereof.

# 1 About this Guide

## 1.1 Introduction

SUSE® Linux Enterprise Server for SAP Applications is the optimal platform to run SAP\* applications with high availability. Together with a redundant layout of the technical infrastructure, single points of failure can be eliminated.

SAP\* Business Suite is a sophisticated application platform for large enterprises and mid-size companies. Many critical business environments require the highest possible SAP\* application availability.

The described cluster solution could be used for SAP\* S/4 HANA as well as for SAP\* SAP NetWeaver.

SAP NetWeaver is a common stack of middleware functionality used to support the SAP business applications. The SAP Enqueue Replication Server constitutes application level redundancy for one of the most crucial components of the SAP NetWeaver stack, the enqueue service. An optimal effect of the enqueue replication mechanism can be achieved when combining the application level redundancy with a high availability cluster solution e.g., as provided by SUSE Linux Enterprise Server for SAP Applications. The described concept has proven its maturity over several years of productive operations for customers of different sizes and branches.

## 1.2 Additional Documentation and Resources

Chapters in this manual contain links to additional documentation resources that are either available on the system or on the Internet.

For the latest documentation updates, see <http://www.suse.com/documentation>.

You can also find numerous whitepapers, a best-practices guide, and other resources at the SUSE Linux Enterprise Server for SAP Applications resource library: <https://www.suse.com/products/sles-for-sap/resource-library/>.

This guide and other SAP specific best practices could be downloaded via <https://www.suse.com/products/sles-for-sap/resource-library/sap-best-practices/>. You can find at this landing page guides for SAP HANA system replication automation and HA scenarios for SAP NetWeaver and SAP S/4 HANA.

## 1.3 Feedback

Several feedback channels are available:

### Bugs and Enhancement Requests

For services and support options available for your product, refer to <http://www.suse.com/support/>.

To report bugs for a product component, go to <https://scc.suse.com/support/> requests, log in, and select Submit New SR (Service Request).

### User Comments

We want to hear your comments about and suggestions for this manual and the other documentation included with this product. Use the User Comments feature at the bottom of each page in the online documentation or go to <http://www.suse.com/doc/feedback> and enter your comments there.

### Mail

For feedback on the documentation of this product, you can also send a mail to [doc-team@suse.de](mailto:doc-team@suse.de) (<mailto:doc-team@suse.de>). Make sure to include the document title, the product version and the publication date of the documentation. To report errors or suggest enhancements, provide a concise description of the problem and refer to the respective section number and page (or URL).

## 2 Scope of this Document

This guide will show you how to:

- Plan a SUSE Linux Enterprise High Availability platform for SAP NetWeaver, including SAP Enqueue Replication Server.
- Set up a Linux high availability platform and perform a basic SAP NetWeaver installation including SAP Enqueue Replication Server on SUSE Linux Enterprise.
- Integrate the high availability cluster with the SAP control framework via `sap-suse-cluster-connector`, as certified by SAP.

This guide focuses on the high availability of the central services. HA cluster solutions for the database and SAP NetWeaver instances are described in the best practice "Simple Stack" available on our landing page (see section "Additional documentation and resources"). For SAP HANA system replication please follow the guides for the performance- or cost-optimized scenario.

## 3 Overview

This guide describes how to set up a pacemaker cluster using SUSE Linux Enterprise Server for SAP Applications 12 for the Enqueue Replication scenario. The goal is to match the SAP NW-HA-CLU 7.40 certification specifications and goals.

These goals include:

- Integration of the cluster with the SAP start framework *sapstartsrv* to ensure that maintenance procedures do not break the cluster stability
- Rolling Kernel Switch (RKS) awareness
- Standard SAP installation to improve support processes

The updated certification SAP NW-HA-CLU 7.40 has redefined some of the test procedures and described new expectations how the cluster should behave in special conditions. These changes allowed us to improve the cluster architecture and to design it for easier usage and setup.

Shared SAP resources are on a central NFS server.

The SAP instances themselves are installed on a shared disk to allow switching over the file systems for proper functionality. The second need for a shared disk is that we are using the SBD for the cluster fencing mechanism STONITH.

### 3.1 Differences to previous Cluster Architectures

The concept is different to the old stack with the master-slave architecture. With the new certification we switch to a more simple model with primitives. This means we have on one machine the ASCS with its own resources and on the other machine the ERS with its own resources.

### 3.2 Three Systems for ASCS, ERS, Database and additional SAP Instances

This guide describes the installation of a distributed SAP system on three systems. In this setup only two systems are in the cluster. The database and SAP dialog instances could also be added to the cluster by either adding the third node to the cluster or by installing the database on either of the nodes. However we recommend to install the database on a separate cluster.



## Note

The cluster in this guide only manages the SAP instances ASCS and ERS, because of the focus of the SAP NW-HA-CLU 7.40 certification.

If your database is SAP HANA, we recommend to set up the performance optimized system replication scenario using our automation solution SAPHanaSR. The SAPHanaSR automation should be set up in an own two node cluster. The setup is described in a separate best practice available at our best practice page.

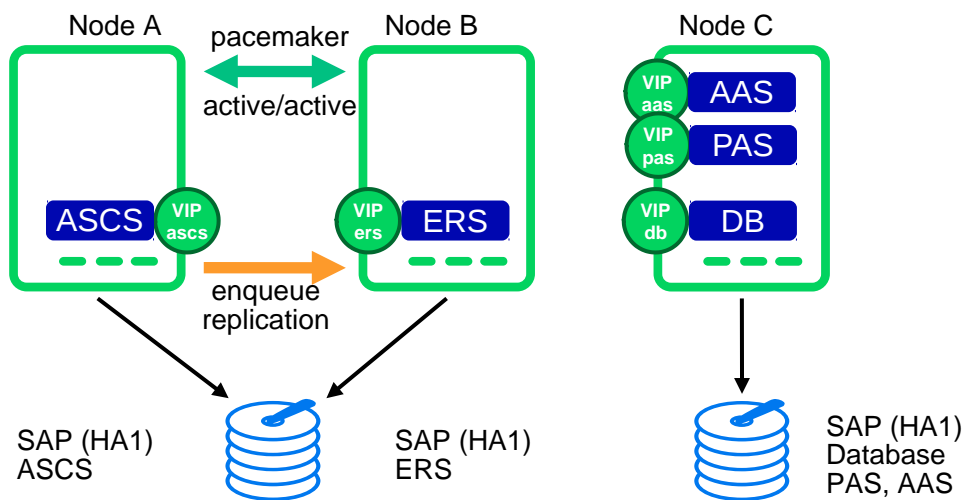


FIGURE 1: THREE SYSTEMS FOR THE CERTIFICATION SETUP

### CLUSTERED MACHINES

- one machine (hacert01) for ASCS
  - Hostname: sapha1as
- one machine (hacert02) for ERS
  - Hostname: sapha1er

### NON-CLUSTERED MACHINE

- one machine (hacert03) for DB and DI

- Hostname: sapha1db
- Hostname: sapha1ci
- Hostname: sapha1d2

### 3.3 High Availability for the Database

Depending on your needs you could also increase the availability of the database, if your database is not already high available by design.

#### 3.3.1 SAP HANA System Replication

A perfect enhancement of the three node scenario described in this document is to implement a SAP HANA system replication (SR) automation.

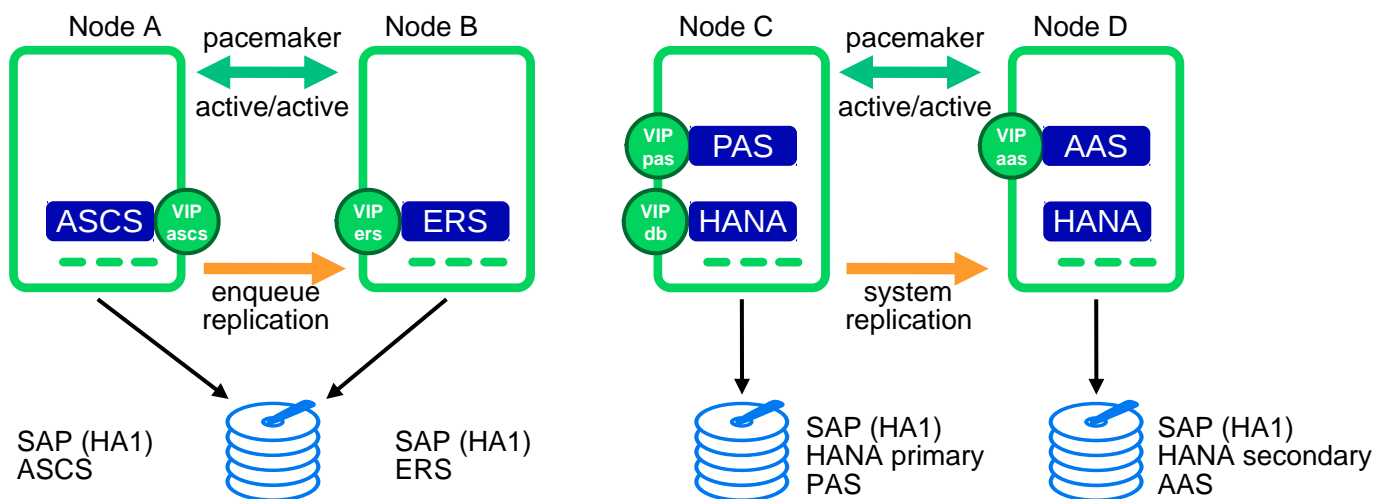


FIGURE 2: ONE CLUSTER FOR CENTRAL SERVICES, ONE FOR SAP HANA SR

The following Databases are supported in combination with this scenario:

- SAP HANA DATABASE 1.0
- SAP HANA DATABASE 2.0

### 3.3.2 Simple Stack

An other option is to implement a second cluster for a database without SR aka "ANYDB". The cluster resource agent SAPDatabase uses the SAPHOSTAGENT to control and monitor the database.

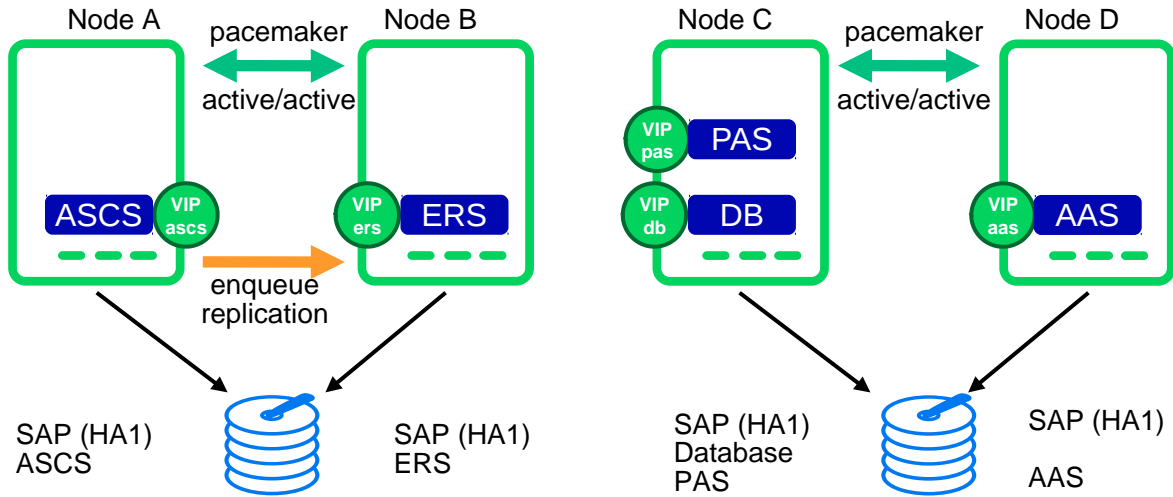


FIGURE 3: ONE CLUSTER FOR THE CENTRAL SERVICES AND ONE CLUSTER FOR THE ANY DATABASE

TABLE 1: THE FOLLOWING OS / DATABASES COMBINATION ARE EXAMPLES FOR THIS SCENARIO

SUSE Linux Enterprise Server for SAP Applications 12	
Intel X86_64	POWER LITTLE ENDIAN
SAP HANA DATABASE 1.0	
SAP HANA DATABASE 2.0	SAP HANA DATABASE 2.0
DB2 FOR LUW 10.5	
MaxDB 7.9	
ORACLE 12.1	
SAP ASE 16.0 FOR BUS. SUITE	





## Note

First version for SAP NetWeaver on Power Little Endian is 7.50. More information about supported combination of OS and Databases for SAP NetWeaver can be found at the SAP Product Availability Matrix. ([SAP PAM \(https://apps.support.sap.com/sap/support/pam\)](https://apps.support.sap.com/sap/support/pam))

### 3.4 Integration of SAP NetWeaver into the Cluster using the Cluster Connector

The integration of the HA cluster through the SAP control framework using the `sap_suse_cluster_connector` is of special interest. The `sapstartsrv` controls SAP instances since SAP Kernel versions 6.40. One of the classical problems running SAP instances in an highly available environment is that if a SAP administrator changes the status (start/stop) of a SAP instance without using the interfaces provided by the cluster software then the cluster framework will detect that as an error status and will bring the SAP instance into the old status by either starting or stopping the SAP instance. This can result in very dangerous situations, if the cluster changes the status of a SAP instance during some SAP maintenance tasks. This new updated solution enables the central component `sapstartsrv` to report state changes to the cluster software, and therefore avoids the previously described dangerous situations. (See also our blog "Using `sap_vendor_cluster_connector` for interaction between cluster framework and `sapstartsrv`") (<https://blogs.sap.com/2014/05/08/using-sapvendorclusterconnector-for-interaction-between-cluster-framework-and-sapstartsrv/comment-page-1/>).

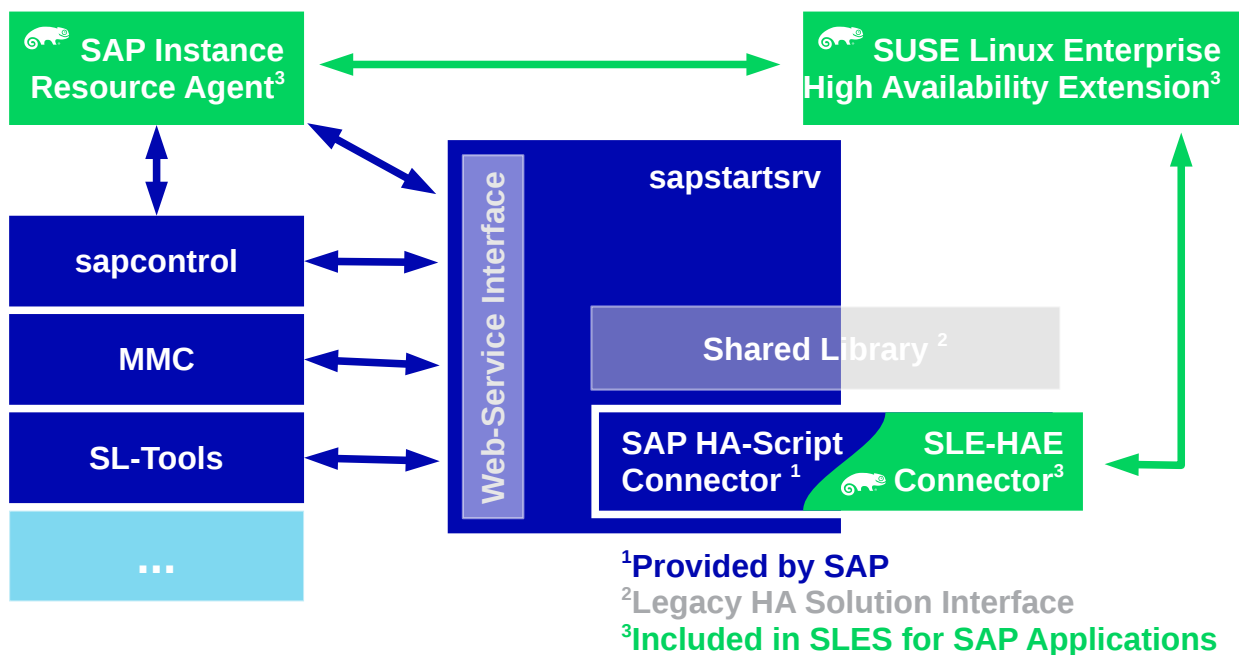


FIGURE 4: CLUSTER CONNECTOR TO INTEGRATE THE CLUSTER WITH THE SAP START FRAMEWORK

### Note

For this scenario we are using an updated version of the sap-suse-cluster-connector which implements the API version 3 for the communication between the cluster framework and the sapstartsrv.

The new version of the sap-suse-cluster-connector now allows to start, stop and *migrate* a SAP instance. The integration between the cluster software and the sapstartsrv also implements to run checks of the HA setup using either the command line tool sapcontrol or even the SAP management consoles (SAP MMC or SAP MC).

## 3.5 Disks and Partitions

For all SAP file systems beside the file systems on NFS we are using XFS.

### 3.5.1 Shared Disk for Cluster ASCS and ERS

The disk for the ASCS and ERS instances need to be shared and assigned to the cluster nodes hacert01 and hacert02. Beside the partitions for the file systems for the SAP instances the disk also provides the partition to be used as SBD.

On hacert01 prepare the file systems for the shared disk. Create three partitions on the shared drive /dev/sdb:

- partition one (/dev/sdb1) for SBD (7M)
- partition two (/dev/sdb2) for the first file system (10GB) formatted with XFS
- partition three (/dev/sdb3) for the second file system (10GB) formatted with XFS

You could either use YaST to create partitions or using available command line tools. The following script could be used for non-interactive setups.

```
# parted -s /dev/sdb print
# # we are on the 'correct' drive, right?
# parted -s /dev/sdb mklabel gpt
# parted -s /dev/sdb mkpart primary 1049k 8388k
# parted -s /dev/sdb mkpart primary 8389k 10.7G
# parted -s /dev/sdb mkpart primary 10.7G 21.5G
# mkfs.xfs /dev/sdb2
# mkfs.xfs /dev/sdb3
```

For these file systems we recommend to use plain partitions to keep the cluster configuration as easy as possible. However you could also place these file systems in separate volume groups. In that case you need to add further cluster resources to control the logical volume groups. This is out of the scope of this setup-guide.

After we have partitioned the shared disk on hacert01 we need to request a partition table rescan on hacert02.

```
# partprobe; fdisk -l /dev/sdb
```

During the SAP installation we need /usr/sap/HA1/ASCS00 to be mounted on hacert01 and /usr/sap/HA1/ERS10 to be mounted on hacert02.

- hacert01: /dev/sdb2 /usr/sap/HA1/ASCS00
- hacert02: /dev/sdb3 /usr/sap/HA1/ERS10

### 3.5.2 Disk for DB and Dialog Instances (MaxDB Example)

The disk for the database and primary application server is assigned to hacert03. In an advanced setup this disk should be shared between hacert03 and an optional additional node building an own cluster.

- partition one (/dev/sdb1) for SBD (7M) - not used here but a reservation for an optional second cluster
- partition two (/dev/sdb2) for the Database (60GB) formatted with XFS
- partition three (/dev/sdb3) for the second file system (10GB) formatted with XFS
- partition four (/dev/sdb4) for the third file system (10GB) formatted with XFS

You could either use YaST to create partitions or using available command line tools. The following script could be used for non-interactive setups.

```
# parted -s /dev/sdb print
# # we are on the 'correct' drive, right?
# parted -s /dev/sdb mklabel gpt
# parted -s /dev/sdb mkpart primary 1049k 8388k
# parted -s /dev/sdb mkpart primary 8389k 60G
# parted -s /dev/sdb mkpart primary 60G 70G
# parted -s /dev/sdb mkpart primary 70G 80G
# mkfs.xfs /dev/sdb2
# mkfs.xfs /dev/sdb3
# mkfs.xfs /dev/sdb4
```

#### TO BE MOUNTED EITHER BY OS OR AN OPTIONAL CLUSTER

- hacert03: /dev/sdb2 /sapdb
- hacert03: /dev/sdb3 /usr/sap/HA1/DVEBMGS01
- hacert03: /dev/sdb4 /usr/sap/HA1/D02



#### Note

DVEBMGS01 ⇒ Since NetWeaver 7.5 the primary application server instance directory has been renamed. (D <Instance\_Number >)

## NFS SERVER

- `nfs1:/data/nfs/suseEnqReplNW740/HA1/sapmnt /sapmnt`
- `nfs1:/data/nfs/suseEnqReplNW740/HA1/usr/sapsys /usr/sap/HA1/SYS`

## MEDIA

- `nfs1:/data/SCT/media/SAP-MEDIA/NW74 /sapcd`

or

- `nfs1:/data/SCT/media/SAP-MEDIA/NW75 /sapcd`

## 3.6 IP Addresses and Virtual Names

Check, if the `/etc/hosts` contains at least the following address resolutions. Add those entries, if they are missing.

```
192.168.201.111 hacert01
192.168.201.112 hacert02
192.168.201.113 hacert03
192.168.201.115 saphalas
192.168.201.116 saphaler
192.168.201.117 saphaldb
192.168.201.118 saphalci
192.168.201.119 saphald2
```

## 3.7 Mount Points and NFS Shares

In our setup the directory `/usr/sap` is part of the root file system. You could of course also create a dedicated file system for that area and mount `/usr/sap` during the system boot. As `/usr/sap` also contains the SAP control file `sapservices` and the `saphostagent` the directory should not be placed on a shared file system between the cluster nodes.

We need to create the directory structure on all nodes which might be able to run the SAP resource. The `SYS` directory will be on a NFS share for all nodes.

- Creating mount points and mounting NFS share at all nodes

### EXAMPLE 1: SAP SAP NETWEAVER 7.4

```
# mkdir -p /sapmnt
```

```
# mkdir -p /usr/sap/HA1/{ASCS00,D02,DVEBMGS01,ERS10,SYS}
# mount -t nfs nfs1:/data/nfs/suseEnqReplNW740/HA1/sapmnt /sapmnt
# mount -t nfs nfs1:/data/nfs/suseEnqReplNW740/HA1/usrsapsys /usr/sap/HA1/SYS
# mount -t nfs nfs1:/data/SCT/media/SAP-MEDIA/NW74 /sapcd
```

#### EXAMPLE 2: SAP SAP NETWEAVER 7.5

```
# mkdir -p /sapmnt
# mkdir -p /usr/sap/HA1/{ASCS00,D01,D02,ERS10,SYS}
# mount -t nfs nfs1:/data/nfs/suseEnqReplNW740/HA1/sapmnt /sapmnt
# mount -t nfs nfs1:/data/nfs/suseEnqReplNW740/HA1/usrsapsys /usr/sap/HA1/SYS
# mount -t nfs nfs1:/data/SCT/media/SAP-MEDIA/NW75 /sapcd
```

- Only MaxDB: creating mount points for the database at hacert03:

```
# mkdir -p /sapdb
```

- Only HANA: creating mount points for database at hacert03:

```
# mkdir -p /hana/{shared,data,log}
```

- Other databases: creating mount points based on there installation guide.

As we do not control the NFS shares via the cluster in this setup, you should add these file systems to */etc/fstab* to get the file systems mounted during the next system boot.

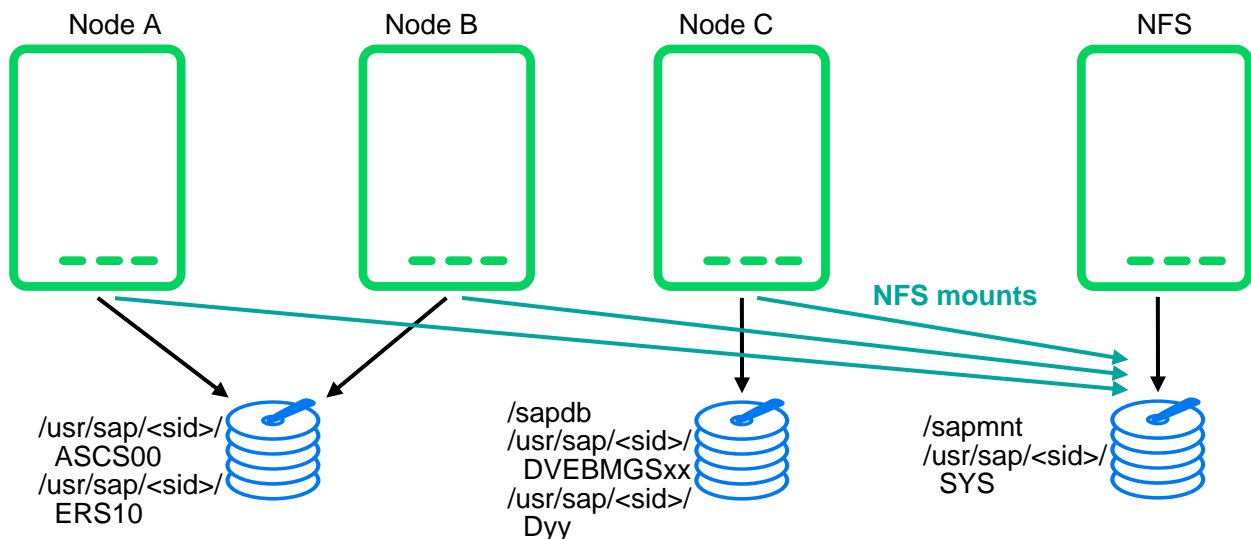


FIGURE 5: FILE SYSTEM LAYOUT INCLUDING NFS SHARES

We prepare the three servers for the distributed SAP installation. Server 1 (hacert01) will be used to install the ASCS SAP instance. Server 2 (hacert02) will be used to install the ERS SAP instance and server 3 (hacert03) will be used to install the dialog SAP instances and the database.

- Mounting the instance and database file systems at one specific node:

#### EXAMPLE 3: SAP NETWEAVER 7.40 ON X86\_64 ARCHITECTURE WITH MAXDB

```
(ASCS hacert01) # mount /dev/sdb2 /usr/sap/HA1/ASCS00
(ERS hacert02) # mount /dev/sdb3 /usr/sap/HA1/ERS10
(DB hacert03) # mount /dev/sdb2 /sapdb
(Dialog hacert03) # mount /dev/sdb3 /usr/sap/HA1/DVEBMGS01
(Dialog hacert03) # mount /dev/sdb4 /usr/sap/HA1/D02
```

#### EXAMPLE 4: SAP NETWEAVER 7.50 ON POWERLE ARCHITECTURE WITH HANA

```
(ASCS hacert01) # mount /dev/sdb2 /usr/sap/HA1/ASCS00
(ERS hacert02) # mount /dev/sdb3 /usr/sap/HA1/ERS10
(DB hacert03) # mount /dev/sdc1 /hana/shared
(DB hacert03) # mount /dev/sdc2 /hana/log
(DB hacert03) # mount /dev/sdc3 /hana/data
(Dialog hacert03) # mount /dev/sdb3 /usr/sap/HA1/D01
(Dialog hacert03) # mount /dev/sdb4 /usr/sap/HA1/D02
```

- As a result the directory `/usr/sap/HA1/` should now look like:

```
# ls -l /usr/sap/HA1/
total 0
drwxr-xr-x 1 haladm sapsys 70 28. Mar 17:26 ./
drwxr-xr-x 1 root sapsys 58 28. Mar 16:49 ../
drwxr-xr-x 7 haladm sapsys 58 28. Mar 16:49 ASCS00/
drwxr-xr-x 1 haladm sapsys 0 28. Mar 15:59 D02/
drwxr-xr-x 1 haladm sapsys 0 28. Mar 15:59 DVEBMGS01/
drwxr-xr-x 1 haladm sapsys 0 28. Mar 15:59 ERS10/
drwxr-xr-x 5 haladm sapsys 87 28. Mar 17:21 SYS/
```



### Note

The owner of the folder and files are changed during the SAP installation. By default all of them are owned by root.

## 4 SAP Installation

The overall procedure to install the distributed SAP is:

- Installing the ASCS instance for the central services
- Installing the ERS to get a replicated enqueue scenario
- Prepare the ASCS and ERS installations for the cluster take-over
- Installing the Database
- Installing the primary application server instance (PAS)
- Installing additional application server instances (AAS)

The result will be a distributed SAP installation as illustrated here:

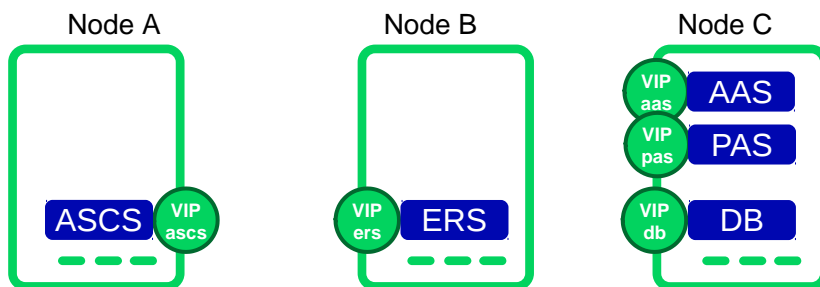


FIGURE 6: DISTRIBUTED INSTALLATION OF THE SAP SYSTEM

### 4.1 Linux User and Group Number Scheme

Whenever asked by the SAP software provisioning manager (SWPM) which Linux User IDs or Group IDs to use, refer to the following table which is, of course, only an example.

Group	sapinst	1000
Group	sapsys	1001
Group	sapadm	3000
Group	sdba	3002
User	ha1adm	3000
User	sdb	3002
User	sqdha1	3003
User	sapadm	3004
User	h04adm	4001



## 4.2 Install ASCS on hacert01

Temporarily we have to set the service IP address we will have later in the cluster, as local IP because the installer would like to resolve or use it. Please make sure to use the right virtual hostname for each installation step. Take care for the file systems like `/dev/sdb2` and `/sapcd/` which might also need to be mounted.

```
# ip a a 192.168.201.115/24 dev eth0
# mount /dev/sdb2 /usr/sap/HA1/ASCS00
# cd /sapcd/SWPM/
# ./sapinst SAPINST_USE_HOSTNAME=sapha1as
```

- SWPM option depends on SAP NetWeaver version and architecture
  - Installing SAP NetWeaver 7.40 SR2 → MaxDB → SAP-Systems → Application Server ABAP → High-Availability System → ASCS Instance
  - Installing SAP NetWeaver 7.5 → SAP HANA Database → Installation → Application Server ABAP → High-Availability System → ASCS Instance
- SID id HA1
- Use instance number 00
- Deselect using FQDN
- All passwords: please use SuSE1234
- Double-check during the parameter review, if virtual name **sapha1as** is used

## 4.3 Install ERS on hacert02

Temporarily we have to set the service IP address we will have later in the cluster, as local IP because the installer would like to resolve or use it. Please make sure to use the right virtual hostname for each installation step.

```
# ip a a 192.168.201.116/24 dev eth0
# mount /dev/sdb3 /usr/sap/HA1/ERS10
# cd /sapcd/SWPM/
# ./sapinst SAPINST_USE_HOSTNAME=sapha1er
```

- SWPM option depends on SAP NetWeaver version and architecture

- Installing SAP NetWeaver 7.40 SR2 → MaxDB → SAP-Systems → Application Server ABAP → High-Availability System → Enqueue Replication Server Instance
- Installing SAP NetWeaver 7.5 → SAP HANA Database → Installation → Application Server ABAP → High-Availability System → Enqueue Replication Server Instance
- Use instance number 10
- Deselect using FQDN
- Double-check during the parameter review, if virtual name **sapha1er** is used
- If you get an error during the installation about permissions, change the ownership of the ERS directory

```
# chown -R haladm:sapsys /usr/sap/HA1/ERS10
```

- If you get a prompt to manually stop/start the ASCS instance, please login at hacert01 as user haladm and call sapcontrol.

```
# sapcontrol -nr 00 -function Stop    # to stop the ASCS
# sapcontrol -nr 00 -function Start   # to start the ASCS
```

## 4.4 Poststeps for ASCS and ERS

### 4.4.1 Stopping ASCS and ERS

*On hacert01*

```
# su - haladm
# sapcontrol -nr 00 -function Stop
# sapcontrol -nr 00 -function StopService
```

*On hacert02*

```
# su - haladm
# sapcontrol -nr 10 -function Stop
# sapcontrol -nr 10 -function StopService
```

## 4.4.2 Maintaining *sapservices*

Ensure `/usr/sap/sapservices` holds both entries (ASCS + ERS) on both cluster nodes. This allows the `sapstartsrv` clients to start the service like

*As user `ha1adm`*

```
# sapcontrol -nr 10 -function StartService HA1
```

The `/usr/sap/sapservices` looks like (typically one line per instance):

```
#!/bin/sh
LD_LIBRARY_PATH=/usr/sap/HA1/ASCS00/exe:$LD_LIBRARY_PATH; export LD_LIBRARY_PATH; /usr/
sap/HA1/ASCS00/exe/sapstartsrv pf=/usr/sap/HA1/SYS/profile/HA1_ASCS00_sapha1as -D -u
ha1adm
LD_LIBRARY_PATH=/usr/sap/HA1/ERS10/exe:$LD_LIBRARY_PATH; export LD_LIBRARY_PATH; /usr/
sap/HA1/ERS10/exe/sapstartsrv pf=/usr/sap/HA1/ERS10/profile/HA1_ERS10_sapha1er -D -u
ha1adm
```

## 4.4.3 Integrating the Cluster Framework using *sap-suse-cluster-connector*

Install the package **sap-suse-cluster-connector** version 3.0.0 from our repositories:

```
# zypper in sap-suse-cluster-connector
```



### Note

Be careful there are two packages available. The package `sap_suse_cluster_connector` continues to contain the old version 1.1.0 (SAP API 1). The package `sap-suse-cluster-connector` contains the new version 3.0.x (SAP API 3). The package `sap-suse-cluster-connector` with version 3.0.x implements the SUSE SAP API version 3. New features like SAP Rolling Kernel Switch (RKS) and migration of ASCS are only supported with this new version.

For the ERS and ASCS instance edit the instance profiles `HA1_ASCS00_sapha1as` and `HA1_ERS10_sapha1er` in the profile directory `/usr/sap/HA1/SYS/profile/`.

You need to tell the `sapstartsrv` to load the HA script connector library and to use the `sap-suse-cluster-connector`.

```
service/halib = $(DIR_CT_RUN)/saphascriptco.so
```

```
service/halib_cluster_connector = /usr/bin/sap_suse_cluster_connector
```

Add the user haladm to the unix user group haclient.

```
# usermod -a -G haclient haladm
```

#### 4.4.4 Adapting SAP Profiles to match the SAP NW-HA-CLU 7.40 Certification

For the ASCS, change the start command from *Restart\_Programm\_xx* to *Start\_Programm\_xx* for the enqueue server (enserver). This change tells the SAP start framework **not** to self-restart the enqueue process. Such a restart would lead in loss of the locks.

File `/usr/sap/HA1/SYS/profile/HA1_ASCS00_sapha1as`.

```
Start_Program_01 = local $_EN pf=$_PF
```

Optionally you could limit the number of restarts of services (in the case of ASCS this limits the restart of the message server).

For the ERS change instance the start command from *Restart\_Programm\_xx* to *Start\_Programm\_xx* for the enqueue replication server (enrepserver).

File `/usr/sap/HA1/SYS/profile/HA1_ERS10_sapha1er`.

```
Start_Program_00 = local $_ER pf=$_PFL NR=$(SCSID)
```

#### 4.4.5 Starting ASCS and ERS

*On hacert01*

```
# su - haladm  
# sapcontrol -nr 00 -function StartService HA1  
# sapcontrol -nr 00 -function Start
```

*On hacert02*

```
# su - haladm  
# sapcontrol -nr 10 -function StartService HA1  
# sapcontrol -nr 10 -function Start
```

## 4.5 Install DB on hacert03 (Example MaxDB)

The MaxDB needs min.40 GB. We use /dev/sdb2 and mount the partition to /sapdb.

```
# ip a a 192.168.201.118/24 dev eth0
# mount /dev/sdb2 /sapdb
# cd /sapcd/SWPM/
# ./sapinst SAPINST_USE_HOSTNAME=sapha1db
```

- We are installing SAP NetWeaver 7.40 SR2 → MaxDB → SAP-Systems → Application Server ABAP → High Availability System → DB
- Profile directory /sapmnt/HA1/profile
- DB ID is HA1
- Volume Media Type **keep** File (not raw)
- Deselect using FQDN
- Double-check during the parameter review, if virtual name **sapha1db** is used

## 4.6 Install DB on hacert03 (Example SAP HANA)

The HANA DB has very strict HW requirements. The storage sizing depends on many indicators. Please check the supported configurations on [SAP HANA Hardware Directory \(https://www.sap.com/documents/2015/03/74cdb554-5a7c-0010-82c7-eda71af511fa.html\)](https://www.sap.com/documents/2015/03/74cdb554-5a7c-0010-82c7-eda71af511fa.html) and [SAP HANA TDI \(https://www.sap.com/documents/2015/03/74cdb554-5a7c-0010-82c7-eda71af511fa.html\)](https://www.sap.com/documents/2015/03/74cdb554-5a7c-0010-82c7-eda71af511fa.html).

```
# ip a a 192.168.201.118/24 dev eth0
# mount /dev/sdc1 /hana/shared
# mount /dev/sdc2 /hana/log
# mount /dev/sdc3 /hana/data
# cd /sapcd/SWPM/
# ./sapinst SAPINST_USE_HOSTNAME=sapha1db
```

- We are installing SAP NetWeaver 7.5 → SAP HANA Database → Installation → Application Server ABAP → High-Availability System → Database Instance
- Profile directory /sapmnt/HA1/profile
- Deselect using FQDN

- Database parameters enter DBSID is H04; Database Host is sapha1db; Instance Number is 00
- Database System ID enter Instance Number is 00; SAP Mount Directory is /hana/shared
- Account parameters change them in case of custom values needed
- Cleanup select **Yes**, remove operating system users from group'sapinst'....
- Double-check during the parameter review, if virtual name **sapha1db** is used

## 4.7 Install the Primary Application Server (PAS) on hacert03

```
# ip a a 192.168.201.117/24 dev eth0
# mount /dev/sdb3 /usr/sap/HA1/DVEBMGS01
# cd /sapcd/SWPM/
# ./sapinst SAPINST_USE_HOSTNAME=sapha1ci
```

- SWPM option depends on SAP NetWeaver version and architecture
  - Installing SAP NetWeaver 7.40 SR2 → MaxDB → SAP-Systems → Application Server ABAP → High-Availability System → Primary Application Server Instance (PAS)
  - Installing SAP NetWeaver 7.5 → SAP HANA Database → Installation → Application Server ABAP → High-Availability System → Primary Application Server Instance (PAS)
- Use instance number 01
- Deselect using FQDN
- For our hands-on setup use a default secure store key
- Do not install Diagnostic Agent
- No SLD
- Double-check during the parameter review, if virtual name **sapha1ci** is used

## 4.8 Install an Additional Application Server (AAS) on hacert03

```
# ip a a 192.168.201.119/24 dev eth0
```

```
# mount /dev/sdb4 /usr/sap/HA1/D02
# cd /sapcd/SWPM/
# ./sapinst SAPINST_USE_HOSTNAME=sapha1d2
```

- SWPM option depends on SAP NetWeaver version and architecture
  - Installing SAP NetWeaver 7.40 SR2 → MaxDB → SAP-Systems → Application Server ABAP → High-Availability System → Additional Application Server Instance (AAS)
  - Installing SAP NetWeaver 7.5 → SAP HANA Database → Installation → Application Server ABAP → High-Availability System → Additional Application Server Instance (AAS)
- Use instance number 02
- Deselect using FQDN
- Do not install Diagnostic Agent
- Double-check during the parameter review, if virtual name **sapha1d2** is used

## 5 Implement the Cluster

The main procedure to implement the cluster is

- Install the cluster software, if not already done during the installation of the operating system
- Configure the cluster communication framework corosync
- Configure the cluster resource manager
- Configure the cluster resources
- Tune the cluster timing in special for the SBD.



### Note

Before we continue to set up the cluster we first stop all SAP instances, remove the (manual added) IP addresses on the cluster nodes and unmount the file systems which will be controlled by the cluster later.



## Note

The SBD device/partition need to be created in beforehand. In this setup guide we already have reserved partition `/dev/sdb1` for SBD usage.

### TASKS

1. Setup NTP (best with yast2) and enable it
2. Install pattern `ha_sles` on both cluster nodes

```
# zypper in -t pattern ha_sles
```

## 5.1 Configure the Cluster Base

### TASKS

- Install and configure the cluster stack at first machine

You could use either YaST to configure the cluster base or use the interactive command line tool `ha-cluster-init`. The following script could be used for automated setups.

```
# modprobe softdog
# echo "softdog" > /etc/modules-load.d/softdog.conf
# systemctl enable sbd
# ha-cluster-init -y csync2
# ha-cluster-init -y -i eth0 corosync
# ha-cluster-init -y -s /dev/sdb1 sbd
# ha-cluster-init -y cluster
```

- Join the second node

Some preparation steps on the second node.

```
# modprobe softdog
# echo "softdog" > /etc/modules-load.d/softdog.conf
# systemctl enable sbd
# rsync 192.168.201.111:/etc/sysconfig/sbd /etc/sysconfig
```

You could use either YaST to configure the cluster base or use the interactive command line tool `ha-cluster-join`. The following script could be used for automated setups.



```
# ha-cluster-join -y -c 192.168.201.111 -i eth0 csync2
# ha-cluster-join -y -c 192.168.201.111 -i eth0 ssh_merge
# ha-cluster-join -y -c 192.168.201.111 -i eth0 cluster
```

- The `crm_mon -lr` output should look like this:

```
Last updated: Thu May 11 14:25:53 2017          Last change: Thu May 11 14:23:21 2017 by
haladm via crm_resource on hacert01
Stack: corosync
Current DC: hacert01 (version 1.1.13-20.1-6f22ad7) - partition with quorum
2 nodes and 1 resource configured

Online: [ hacert01 hacert02 ]

stonith-sbd      (stonith:external/sbd): Started hacert01
```

## 5.2 Configure Cluster Resources

We need a changed `SAPInstance` resource agent for SAP NetWeaver in order **not** to use the Master-Slave construct anymore and move to a more cluster like construct to start and stop the ASCS and the ERS itself and **not** only the complete master-slave.

For this there is a new functionality for the ASCS needed to follow the ERS. The ASCS needs to mount the shared memory table of the ERS to avoid the loss of locks.

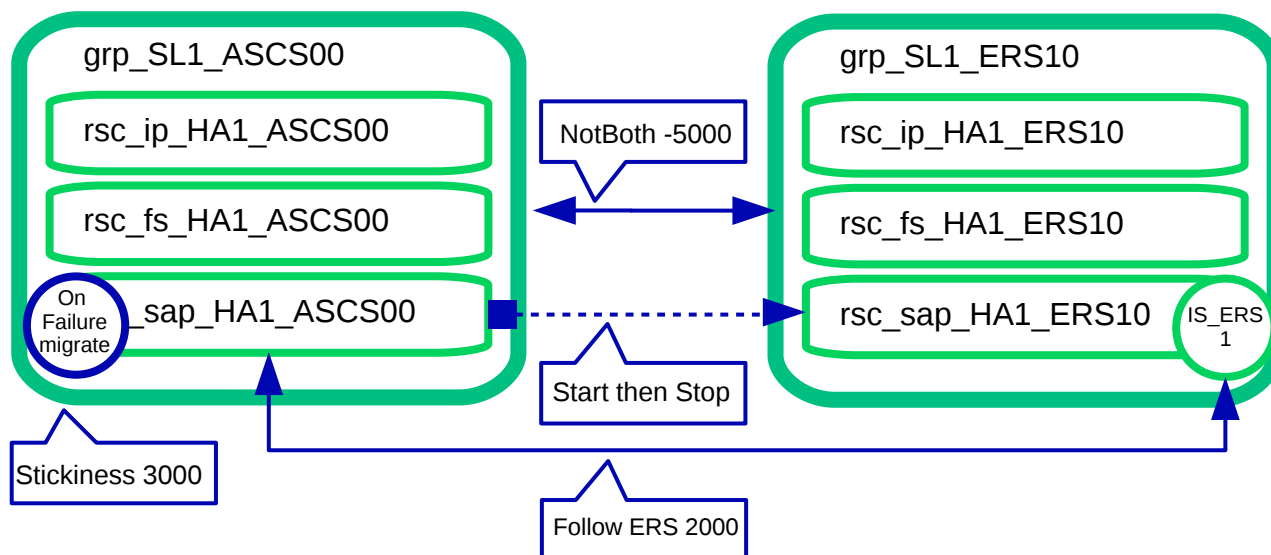


FIGURE 7: RESOURCES AND CONSTRAINTS

The implementation is done with the help of a new flag "runs\_ers\_\$SID" within the RA, enabled with help of the resource parameter "IS\_ERS=TRUE".

Another benefit of this concept is, that we can now work with local (mountable) file systems instead of a shared (NFS) file system for the SAP instance directories.

### 5.2.1 Preparing the Cluster for adding the Resources

To avoid that the cluster starts partially defined resources we set the cluster to the maintenance mode. This deactivates all monitor actions.

*As user root*

```
# crm configure property maintenance-mode="true"
```

### 5.2.2 Configure the Resources for the ASCS

First we configure the resources for the file system, IP address and the SAP instance. Of course you need to adapt the parameters to your environment.

#### EXAMPLE 5: ASCS PRIMITIVE

```
primitive rsc_fs_HA1_ASCS00 Filesystem \  
  params device="/dev/sdb2" directory="/usr/sap/HA1/ASC00" \  
    fstype=trfs \  
  op start timeout=60s interval=0 \  
  op stop timeout=60s interval=0 \  
  op monitor interval=20s timeout=40s  
primitive rsc_ip_HA1_ASCS00 IPAddr2 \  
  params ip=192.168.201.115 \  
  op monitor interval=10s timeout=20s  
primitive rsc_sap_HA1_ASCS00 SAPIInstance \  
  operations $id=rsc_sap_HA1_ASCS00-operations \  
  op monitor interval=11 timeout=60 on_fail=restart \  
  params InstanceName=HA1_ASCS00_saphalas \  
    START_PROFILE="/sapmnt/HA1/profile/HA1_ASCS00_saphalas" \  
    AUTOMATIC_RECOVER=false \  
  meta resource-stickiness=5000 failure-timeout=60 \  
    migration-threshold=1 priority=10
```

#### EXAMPLE 6: ASCS GROUP

```
group grp_HA1_ASCS00 \  
  primitive rsc_fs_HA1_ASCS00 Filesystem \  
  primitive rsc_ip_HA1_ASCS00 IPAddr2 \  
  primitive rsc_sap_HA1_ASCS00 SAPIInstance
```

```
rsc_ip_HA1_ASCS00 rsc_fs_HA1_ASCS00 rsc_sap_HA1_ASCS00 \  
meta resource-stickiness=3000
```

Create a txt file (like `crm_ascs.txt`) with your preferred text editor, enter both examples (primitives and group) to that file and load the configuration to the cluster manager configuration.

*As user root*

```
# crm configure load update crm_ascs.txt
```

### 5.2.3 Configure the Resources for the ERS

Second we configure the resources for the file system, IP address and the SAP instance. Of course you need to adapt the parameters to your environment.

The specific parameter `IS_ERS = true` should only be set for the ERS instance.

#### EXAMPLE 7: ERS PRIMITIVE

```
primitive rsc_fs_HA1_ERS10 Filesystem \  
  params device="/dev/sdb3" directory="/usr/sap/HA1/ERS10" fstype=xfst \  
  op start timeout=60s interval=0 \  
  op stop timeout=60s interval=0 \  
  op monitor interval=20s timeout=40s \  
primitive rsc_ip_HA1_ERS10 IPaddr2 \  
  params ip=192.168.201.116 \  
  op monitor interval=10s timeout=20s \  
primitive rsc_sap_HA1_ERS10 SAPIInstance \  
  operations $id=rsc_sap_HA1_ERS10-operations \  
  op monitor interval=11 timeout=60 on_fail=restart \  
  params InstanceName=HA1_ERS10_saphaler \  
    START_PROFILE="/sapmnt/HA1/profile/HA1_ERS10_saphaler" \  
    AUTOMATIC_RECOVER=false IS_ERS=true \  
  meta priority=1000
```

#### EXAMPLE 8: ERS GROUP

```
group grp_HA1_ERS10 \  
  rsc_ip_HA1_ERS10 rsc_fs_HA1_ERS10 rsc_sap_HA1_ERS10
```

Create a txt file (like `crm_ers.txt`) with your preferred text editor, enter both examples (primitives and group) to that file and load the configuration to the cluster manager configuration.

*As user root*

```
# crm configure load update crm_ers.txt
```

#### 5.2.4 Configure the Colocation Constraints between ASCS and ERS

The constraints between the ASCS and ERS instance are needed to define that the ASCS instance should start-up exactly on the cluster node running the ERS instance after a failure (`loc_sap_HA1_failover_to_ers`). This constraint is needed to ensure that the locks are not lost after a ASCS instance (or node) failure.

If the ASCS instance has been started by the cluster the ERS instance should be moved to an "other" cluster node (`col_sap_HA1_no_both`). This constraint is needed to ensure that the ERS will sync the locks again and the cluster is ready for an additional take-over.

##### EXAMPLE 9: LOCATION CONSTRAINT

```
colocation col_sap_HA1_no_both -5000: grp_HA1_ERS10 grp_HA1_ASCS00
location loc_sap_HA1_failover_to_ers rsc_sap_HA1_ASCS00 \
    rule 2000: runs_ers_HA1 eq 1
order ord_sap_HA1_first_start_ascs Optional: rsc_sap_HA1_ASCS00:start \
    rsc_sap_HA1_ERS10:stop symmetrical=false
```

Create a txt file (like `crm_col.txt`) with your preferred text editor, enter all three constraints to that file and load the configuration to the cluster manager configuration.

*As user root*

```
# crm configure load update crm_col.txt
```

#### 5.2.5 Activating the Cluster

Now the last step is to end the cluster maintenance mode and to allow the cluster to detect already running resources.

*As user root*

```
# crm configure property maintenance-mode="false"
```

## 6 Administration

### 6.1 Do and Don't Do

#### 6.1.1 Never stop the ASCS Instance

For normal operation **do not stop** the ASCS SAP instance with any tool such as cluster tools or SAP tools. The stop of the ASCS instance might lead to a loss of enqueue locks. Because following the new SAP NW-HA-CLU 7.40 certification the cluster must allow local restarts of the ASCS. This feature is needed to allow rolling kernel switch (RKS) updates without reconfiguring the cluster.



#### Warning

Stopping the ASCS instance might lead into the loss of SAP enqueue locks during the start of the ASCS on the same node.

#### 6.1.2 How to migrate ASCS

To **migrate** the ASCS SAP instance you should use the SAP tools such as the SAP management console. This will trigger sapstartsrv to use the sap-suse-cluster-connector to migrate the ASCS instance. As user *ha1adm* you might call the following command to migrate-away the ASCS. The migrate-away will always migrate the ASCS to the ERS side which will keep the SAP enqueue locks.

As *ha1adm*

```
# sapcontrol -nr 00 -function HAFailoverToNode ""
```

#### 6.1.3 Never Block Resources

With SAP NW-HA-CLU 7.40 it is **not longer allowed to block resources** from being controlled manually. This using the variable *BLOCK\_RESOURCES* in */etc/sysconfig/sap\_suse\_cluster\_connector* is not allowed any more.

#### 6.1.4 Always use unique Instance Numbers

Currently all SAP **instance numbers controlled by the cluster must be unique**. If you need to have multiple dialog instances such as D00 running on different systems they should be not controlled by the cluster.

#### 6.1.5 How to set Cluster in Maintenance Mode

Procedure to set the cluster into maintenance mode can be done as *root* or *sidadm*.

*As user root*

```
# crm configure properties maintenance-mode="true"
```

*As user ha1adm (the full path is needed)*

```
# /usr/sbin/crm configure properties maintenance-mode="true"
```

#### 6.1.6 Procedure to end the Cluster Maintenance

*As user root*

```
# crm configure properties maintenance-mode="false"
```

#### 6.1.7 Cleanup Resources

How to **cleanup resource failures**? Failures of the ASCS will be automatically deleted to allow a failback after the configured period of time. For all other resources you can cleanup the status including the failures:

*As user root*

```
# crm resource cleanup RESOURCE-NAME
```



## Warning

You should not cleanup the complete group of the ASCS resource as this might lead into an unwanted cluster action to take-over the complete group to the node where ERS instance is running.

## 6.2 Testing the Cluster

We strongly recommend that you at least process the following tests before you plan going into production with your cluster:

### 6.2.1 Check Product Names with HAGetFailoverConfig

Check if the name of the SUSE cluster solution is shown in the output of sapcontrol or SAP management console. This test checks the status of the SAP NetWeaver cluster integration.

*As user ha1adm*

```
# sapcontrol -nr 00 -function HAGetFailoverConfig
```

### 6.2.2 Start SAP Checks using HACheckConfig and HACheckFailoverConfig

Check if the HA configuration tests are showing no errors.

*As user ha1adm*

```
# sapcontrol -nr 00 -function HACheckConfig  
# sapcontrol -nr 00 -function HACheckFailoverConfig
```

### 6.2.3 Manually migrate ASCS

Check if manually migrating the ASCS using HA tools works properly

*As user root*

```
# crm resource migrate rsc_sap_HA1_ASCS00 force
```

```
## wait till the ASCS is been migrated to the ERS host
# crm resource unmigrate rsc_sap_HA1_ASCS00
```

## 6.2.4 Migrate ASCS using HAFailoverToNode

Check if moving the ASCS instance using SAP tools like sapcontrol does work properly

*As user ha1adm*

```
# sapcontrol -nr 00 -function HAFailoverToNode ""
```

## 6.2.5 Test ASCS migration after Failure

Check if the ASCS instance moves correctly after a node failure

*As user root*

```
## on the ASCS host
# echo b >/proc/sysrq-trigger
```

## 6.2.6 Inplace restart of ASCS using Stop and Start

Check if the inplace re-start of the SAP resources have been processed correctly. The SAP instance should not failover to an other node, it must start on the same node where it has been stopped.



### Warning

This test will force the SAP system to **lose** the enqueue locks. **This test should not be processed during production.**

*As user ha1adm*

```
## example for ASCS
# sapcontrol -nr 00 -function Stop
## wait till the ASCS is completly down
# sapcontrol -nr 00 -function Start
```



## 6.2.7 Additionally you should test

- Automated restart of the ASCS (simulating RKS)
- Check the recoverable and non-recoverable outage of the message server process
- Check the non-recoverable outage of the SAP enqueue server process
- Check the outage of the SAP Enqueue Replication Server
- Check the outage and restart of sapstartsrv
- Check the rolling kernel switch procedure (RKS), if possible
- Check the simulation of an upgrade
- Check the simulation of cluster resource failures

# 7 Appendix

## 7.1 CRM Config

The complete crm config for SAP system HA1

```
#
# nodes
#
node 1084753931: hacert01
node 1084753932: hacert02
#
# primitives for ASCS and ERS
#
primitive rsc_fs_HA1_ASCS00 Filesystem \
    params device="/dev/sdb2" directory="/usr/sap/HA1/ASCS00" \
    fstype=xfs \
    op start timeout=60s interval=0 \
    op stop timeout=60s interval=0 \
    op monitor interval=20s timeout=40s
primitive rsc_fs_HA1_ERS10 Filesystem \
    params device="/dev/sdb3" directory="/usr/sap/HA1/ERS10" \
    fstype=xfs \
    op start timeout=60s interval=0 \
    op stop timeout=60s interval=0 \
    op monitor interval=20s timeout=40s
```

```

primitive rsc_ip_HA1_ASCS00 IPAddr2 \
    params ip=192.168.201.115 \
    op monitor interval=10s timeout=20s
primitive rsc_ip_HA1_ERS10 IPAddr2 \
    params ip=192.168.201.116 \
    op monitor interval=10s timeout=20s
primitive rsc_sap_HA1_ASCS00 SAPIInstance \
    operations $id=rsc_sap_HA1_ASCS00-operations \
    op monitor interval=11 timeout=60 on_fail=restart \
    params InstanceName=HA1_ASCS00_saphalas \
    START_PROFILE="/sapmnt/HA1/profile/HA1_ASCS00_saphalas" \
    AUTOMATIC_RECOVER=false \
    meta resource-stickiness=5000 failure-timeout=60 migration-threshold=1 \
    priority=10
primitive rsc_sap_HA1_ERS10 SAPIInstance \
    operations $id=rsc_sap_HA1_ERS10-operations \
    op monitor interval=11 timeout=60 on_fail=restart \
    params InstanceName=HA1_ERS10_saphaler \
    START_PROFILE="/sapmnt/HA1/profile/HA1_ERS10_saphaler" \
    AUTOMATIC_RECOVER=false IS_ERS=true \
    meta priority=1000
primitive stonith-sbd stonith:external/sbd \
    params pcmk_delay_max=30s
#
# group definitions for ASCS and ERS
#
group grp_HA1_ASCS00 rsc_ip_HA1_ASCS00 rsc_fs_HA1_ASCS00 rsc_sap_HA1_ASCS00 \
    meta resource-stickiness=3000
group grp_HA1_ERS10 rsc_ip_HA1_ERS10 rsc_fs_HA1_ERS10 rsc_sap_HA1_ERS10
#
# constraints between ASCS and ERS
#
colocation col_sap_HA1_not_both -5000: grp_HA1_ERS10 grp_HA1_ASCS00
location loc_sap_HA1_failover_to_ers rsc_sap_HA1_ASCS00 \
    rule 2000: runs_ers_HA1 eq 1
order ord_sap_HA1_first_ascs Optional: rsc_sap_HA1_ASCS00:start rsc_sap_HA1_ERS10:stop
symmetrical=false
#
# crm properties and more
#
property cib-bootstrap-options: \
    have-watchdog=true \
    dc-version=1.1.13-20.1-6f22ad7 \
    cluster-infrastructure=corosync \
    cluster-name=hacluster \
    stonith-enabled=true \
    placement-strategy=balanced \

```

```
last-lrm-refresh=1494346532
rsc_defaults rsc-options: \
  resource-stickiness=1 \
  migration-threshold=3
op_defaults op-options: \
  timeout=600 \
  record-pending=true
```

## 7.2 Related SAP Notes

- 953653 - Rolling Kernel Switch (<https://launchpad.support.sap.com/#/notes/953653/E>)
- 1092448 - IBM XL C/C++ runtime environment for Linux on system p (<https://launchpad.support.sap.com/#/notes/1092448/E>)
- 1153713 - Problems with SAP Management Console (Java) (<https://launchpad.support.sap.com/#/notes/1153713/E>)
- 1763512 - Support details for SUSE Linux Enterprise for SAP Applications (<https://launchpad.support.sap.com/#/notes/1763512/E>)
- 1984787 - SUSE LINUX Enterprise Server 12: Installation notes (<https://launchpad.support.sap.com/#/notes/1984787/E>)
- 2077934 - Rolling kernel switch in HA environments (<https://launchpad.support.sap.com/#/notes/2077934/E>)
- 2235581 - SAP HANA: Supported Operating Systems (<https://launchpad.support.sap.com/#/notes/2235581/E>)
- 2254173 - Linux: Rolling Kernel Switch in Pacemaker based NetWeaver HA environments (<https://launchpad.support.sap.com/#/notes/2254173/E>)
- 2369910 - SAP Software on Linux: General information (<https://launchpad.support.sap.com/#/notes/2369910/E>)