SAP NetWeaver High Availability Cluster 7.40 for the AWS Cloud - Setup Guide

Fabian Herschel, Bernd Schubert, Stefan Schneider (AWS)

Publication Date: 2018/7/4

Contents

Revision 1.1 from 2018-07-04

SUSE LLC

10 Canal Park Drive

Suite 200

Cambridge MA 02142

USA

http://www.suse.com/documentation

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). 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.
- Install HA cluster solutions for the SAP HANA database on AWS as being described in SAP note 2309342 (SUSE Linux Enterprise High Availability Extension on AWS).

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 on the AWS platform. This guide does not document how to install on premises pacemaker clusters. 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 being managed in AWS Elastic File Systems (EFS). The SAP instances themselves are installed on EFS file systems to allow switching over the file systems for proper functionality.

3.1 Using AWS Architectures in SLES Pacemaker Clusters

SLES pacemaker clusters will be installed in an AWS region. An AWS region consists of multiple availability zones. Availability zones are located in different data centers which are 10 to 50km apart. Availability zones have independent flood levels, electricity and network hookup. They are supposed to be independent. AWS recommends architectural patterns where redundant cluster nodes are being spread across availability zones (AZs) in order to allow a customer to overcome individual AZ failures.

An AWS Virtual Private Network (VPC) is spanning all AZs. We assume that a customer will have.

- Identified two availability zones to be used
- Created subnets in the two AZs which can host the two nodes of a SLES HAE cluster

- Use a routing table which is attached to the two subnets
- Optionally: host a Route53 private hosted naming zone to manage names in the VPC

The AWS specific components can be installed in two configurations. Both configurations use the AWS Overlay IP address. An Overlay IP address is an AWS specific routing entry which can send network traffic to an instance, no matter which AZ the instance is located in.

The SLES HAE cluster will update this routing entry as it is required. All SAP system components in the VPC will be able to reach an AWS instance with a SAP system component inside a VPC through this Overlay IP address.

Overlay IP addresses have one disadvantage: they have to come from a CIDR range which is outside of the VPC. Otherwise they would be part of a subnet and a given availability zone.

On premises users like SAP GUIs will not be able to reach this IP address since the AWS Virtual Private Network (VPN) gateway will not route traffic to such an IP address. A customer has two options to overcome this limitation.

1. Use a SAP Router in the VPC. On premises users can reach it. The SAP router can relay traffic to the ASCS system.
2. Configure the additional Route 53 agent. Route 53 is the AWS specific name service. The cluster agent will change the IP address for a given name of the ASCS service. The on premises name server will have to delegate requests to the sub domain in the AWS VPC to this name service. On premises SAP GUI user will contact the ASCS through a name. Section TBD in the appendix explains how to integrate Route 53 with your local naming services.

3.2 Prerequisites for the AWS specific HA Installation

There are a number of prerequisites which have to be met before starting the installation:

- Have an AWS account
- Have an AWS user with admin rights. At least rights to
  - Create security groups
  - Create EFS file systems
  - Modify AWS routing tables

    - Create policies and attach them to IAM roles
    - Optional for Route53 agent installation
        - Create and modify A-records in a private hosted zone
- Understand your landscape
    - Know your region and it’s AWS name
    - Know your VPC and it’s AWS id
    - Know which availability zones you want to use in your VPC
    - Have a subnet in each of the availability zones
        - Have a routing table which is implicitly or explicitly attached to the two subnets
        - Have free IP addresses in the two subnets for your SAP installation and EFS mount points
        - Allow network traffic in between the two subnets
        - Allow outgoing Internet access from the subnets
    - Optionally: Have a Route 53 private hosted zone which hosts a subdomain for instances in the two subnets
    - Have a resource record with a name and the IP address for the SAP central instance

Please use the check list in the appendix to note down all information needed before starting the installation.

3.2.1 Tagging the EC2 Instances

The EC2 instances will have host names which are automatically generated. Select host names which comply with SAP requirements, see SAP note 611361.

The cluster agents will have to be able to identify the EC2 instances in the correct way. This happens through instance tags.

Tag the two EC2 instances through the console or the AWS Command Line Interface (CLI) with arbitrarily chosen tag like *cluster* and the host name as it will be shown in the command *uname* .

Use the same tag (like *cluster*) and the individual host names for both instances. The AWS documentation (http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Using_Tags.html ↗) explains how to tag EC2 instances.

Note

Refrain from using non ASCII characters in any tag assigned to cluster managed resources.

3.2.2 Security Groups

This section does not cover a discussion of SAP related ports in security groups. This section lists the ports which need to be available for the SUSE cluster only.

The following ports and protocols need to be configured to allow the two cluster nodes to communicate with each other:

- Port 5405 for inbound UDP: Used to configure the corosync communication layer. Port 5405 is being used in common examples. A different port may be used depending on the corosync configuration.
- Port 7630 for inbound TCP: Used by the SUSE "hawk" web GUI.
- enable ICMP: Used through a ping command in the AWS IP-move agent of the SUSE cluster.

We assume that there are no restriction for outbound network communication.

3.2.3 Creating an AWS CLI Profile on both EC2 Instances

The SLES agents use the AWS Command Line Interface (CLI). They will use an AWS CLI profile which needs to be created for the root account *root* on both instances. The SUSE resources require a profile which creates output in the text format. The name of the profile is arbitrary. The name chosen in this example is *cluster*. The region of the instance needs to be added as well. Replace the string *region-name* with your target region in the following example.

One way to create such a profile is to create a file */root/.aws/config* with the following content:

```
[default]
region = region-name
```

```
[profile cluster]
region = region-name
output = text
```

The other way is to use the *aws configure* CLI command in the following way:

```
# aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]: region-name
Default output format [None]:

# aws configure --profile cluster
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]: region-name
Default output format [None]: text
```

This command sequence generates a default profile and a cluster profile.

3.2.4 Configure http Proxies

This action is not needed if the system has transparent access to the Internet. The resource agents execute AWS CLI (Command Line Interface) commands. These commands send http/https requests to an access point in the Internet. These access point are usually directly reachable. Systems which don't offer transparent Internet access will have to provide a http/https proxy. The configuration of the proxy access is described in full detail in the AWS documentation.

Please add the following environment variables to the root user's *.bashrc* file:

```
export HTTP_PROXY=http://a.b.c.d:n
export HTTPS_PROXY=http://a.b.c.d:m
```

Please add the following environment variables instead of the ones above if authentication is required:

```
export HTTP_PROXY=http://username:password@a.b.c.d:n
export HTTPS_PROXY=http://username:password@a.b.c.d:m
```

The AWS Data Provider for SAP will need to reach the instance meta data service directly. Please add the following environment variable to the root user's *.bashrc* file:

```
export NO_PROXY=169.254.169.254
```

3.2.5 Disable the Source/Destination Check for the Cluster Instances

The source/destination check can be disabled through scripts using the AWS command line interface (AWS-CLI). The following command needs to be executed one time for both EC2 instances, which are supposed to receive traffic from the Overlay IP address:

```
# aws ec2 modify-instance-attribute --profile cluster --instance-id EC2-instance --no-
source-dest-check
```

The system on which this command gets executed needs temporarily a role with the following policy:

```
{
   "Version": "2012-10-17",
   "Statement": [
   {
      "Sid": "Stmt1424870324000",
      "Effect": "Allow",
      "Action": [ "ec2:ModifyInstanceAttribute"],
      "Resource": [
      "arn:aws:ec2:region-name:account-id:instance/instance-a",
      "arn:aws:ec2:region-name:account-id:instance/instance-b"
      ]
   }
   ]
}
```

Replace the individual parameter for the region, the account identifier and the two identifiers for the EC2 instances with appropriate values.

The source/destination check can be disabled as well from the AWS console.

3.2.6 AWS Roles and Policies

The SAP ASCS and ESR will run the SLES Pacemaker software and the agents. This software needs a number of AWS IAM privileges to operate the cluster. Create a new Role for every ASCS/ESR cluster and associate this role to the two instances. Attach the following policies to this role:

3.2.6.1 AWS Data Provider Policy

Every cluster node will operate a SAP system. SAP systems on AWS require an installation of the “AWS Data Provider for SAP”. The data provider needs a policy to access AWS resources. Use the policy as described in the “AWS Data Provider for SAP Installation and Operations Guide“ section IAM Roles and attach it to the role of the instance. This policy can be used by all SAP systems. It takes only one policy in an AWS account. The policy doesn’t contain any instance specific privileges.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "EC2:DescribeInstances",
                "EC2:DescribeVolumes"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "cloudwatch:GetMetricStatistics",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::aws-data-provider/config.properties"
        }
    ]
}
```

3.2.6.2 STONITH Policy

The instances of the SUSE cluster will need the privilege to start and stop the other nodes in the cluster. Create a policy with a name like *stonith-policy* with the following content and attach it to the cluster role:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1424870324000",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceAttribute",
                "ec2:DescribeTags"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Stmt1424870324001",
            "Effect": "Allow",
            "Action": [
                "ec2:ModifyInstanceAttribute",
                "ec2:RebootInstances",
                "ec2:StartInstances",
                "ec2:StopInstances"
            ],
            "Resource": [
                "arn:aws:ec2:us-east-1:aws-account:instance/i-node1",
                "arn:aws:ec2:us-east-1:aws-account:instance/i-node2"
            ]
        }
    ]
}
```

Replace the variable *aws-account* with the appropriate AWS account identifier. Replace the variables *i-node1* and *i-node2* with the AWS instance-ids of your two cluster nodes of (hacert01) and (hacert02). This policy is dependent of the instances of your cluster. You will need a separate policy for every cluster!

3.2.7 Overlay IP Agent Policy

The Overlay IP agent will change a routing entry in an AWS routing table. Create a policy with a name like *Manage-Overlay-IP-Policy* and attach it to the role of the cluster instances:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1424870324000",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceAttribute",
                "ec2:DescribeTags"
            ],
            "Resource": "*"
        },
        {
            "Sid": "Stmt1424860166260",
            "Action": [
                "ec2:CreateRoute",
                "ec2:DeleteRoute",
                "ec2:DescribeRouteTables",
                "ec2:ReplaceRoute"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

This policy allows the cluster instances to change any routing. It is currently (Dec. 2017) not possible to narrow down this privilege to individual routing tables. This policy is therefore independent of the cluster and can be reused in the AWS account for other clusters.

3.2.8 Route 53 Updates

It is optional to install the Route 53 agent in the cluster. This policy is needed only when the Route 53 agent will be used. Create a policy with the name *Route53-Update* and attach it to the role of the two cluster nodes:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1471878724000",
            "Effect": "Allow",
            "Action": "route53:GetChange",
            "Resource": "arn:aws:route53:::change/*"
        },
        {
            "Sid": "Stmt1471878724001",
            "Effect": "Allow",
            "Action": "route53:ChangeResourceRecordSets",
            "Resource": "arn:aws:route53:::hostedzone/hosted zone ID/full name"
        },
        {
            "Sid": "Stmt1471878724002",
            "Effect": "Allow",
            "Action": [
                "route53:ListResourceRecordSets",
                "route53:ChangeResourceRecordSets"
            ],
            "Resource": "arn:aws:route53:::hostedzone/hosted zone ID"
        }
    ]
}
```

This policy is specific to the hosted zone and the resource record set. Replace the variables *hosted zone ID* with the AWS id of the hosted zone. Replace *full name* with the name of the entry. An individual Route 53 policy needs to be created individually for every cluster.

3.3 Add Overlay IP Addresses to Routing Table

Manually add two routing entries to the routing table which is assigned to the two subnets. The IP addresses have to be outside of the CIDR range of the VPC. Use the AWS console and search for “VPC”.

- Select VPC
- Click on “Route Tables” in the left column
- Select route table used for SAP ASCS subnets
- Click on tabulator “Routes”
- Click on “Edit”
- Scroll to the end of the list and click on “Add another route”

3.3.1 Add the Service IP Address for your ASCS Service

Add the service IP address of the ASCS service (node hacert01). Use as filter /32 (example: 192.168.10.1/32). Add the Elastic Network Interface (ENI) name of your instance which is initially serving as ASCS. Save your changes by clicking on “Save”.

This is the service IP address with the name sapha1as.

3.3.2 Add the Service IP Address for your ERS Service

Add the service IP address of the ERS service (node hacert02). Use as filter /32 (example: 192.168.10.2/32). Add the Elastic Network Interface (ENI) name of your instance which is initially serving as ERS. Save your changes by clicking on “Save”.

This is the IP address with the name sapha1er.

3.4 EFS File System

The cluster will need an NFS file system being provided by AWS Elastic File System (EFS). The file system will manage

- /usr/sap/HA1 data for ASCS00, ERS10, D02, DVEBMGS01 and the other application servers and the SYS directory
- /sapmnt

You will need the identifier of your VPC and the subnet identifiers of the subnets in which you plan to operate the two cluster nodes. It is Okay to pick other subnets. These subnets have to be reachable by the two cluster nodes and they have to be in the same availability zone (AZ) for high availability reasons. The option “General Purpose” will be sufficient.

Note down the DNS name of your specific EFS server. AWS name services will resolve it internally to your VPC to an IP address which is in your availability zone. We will refer to this name as "efs-name" when we will have to mount the file systems.

We will use one file system for two future mount points (/usr/sap/HA1/ASCS00, /usr/SAP/HA1/ESR10). This keeps the administration level low and it will provide more throughput. AWS throughput in EFS is based on the total size of the file system.

Login into one of the two cluster nodes and create a number of directories in the EFS file system through a temporary mount. Become root and execute the following commands:

```
# mount efs-name: /mnt
# mkdir -p /mnt/ASCS00 /mnt/ERS10 /mnt/D01 /mnt/DVEBMGS01 /mnt/D02 /mnt/SYS /mnt/sapmnt /
mnt/sapcd
# umount /mnt
```

Create additional directories for other application servers. This NFS file system will be used for all of them.

Mount the two mount points in the cluster nodes. Execute the following command on both cluster nodes as root:

```
# mkdir -p /sapmnt /usr/sap/HA1/SYS
```

Add the following two lines to the file /etc/fstab on the two instances which will run the SAP ASCS and the ERS service.

```
efs-name:SYS     /usr/sap/HA1/SYS    nfs4
 rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2    0 0
efs-name:sapmnt  /sapmnt             nfs4
 rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2    0 0
```

Replace *efs-name* with the appropriate DNS name.

Mount the file system as root with the command

```
# mount /usr/sap/HA1/SYS
# mount /sapmnt
```

3.4.1 Enable Cluster Instances to use the Overlay IP Address

The two cluster instances need the Overlay IP address to be configured as secondary IP address on their standard interface *eth0*. This can be achieved by the command:

```
# ip address add OVERLAY-IP dev eth0
```

Execute this command with root privileges on both instances. Add the ASCS IP address on the ASCS node hacert01. Add the Enque Replication address on the ERS node hacert02.

3.5 Differences to previous Cluster Architecture

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.

3.6 Five Systems for ASCS, ERS, Database and additional SAP Instances

This guide describes the installation of a distributed SAP system on the five 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 three nodes 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. https://www.suse.com/products/sles-for-sap/resource-library/sap-best-practices/ ↗

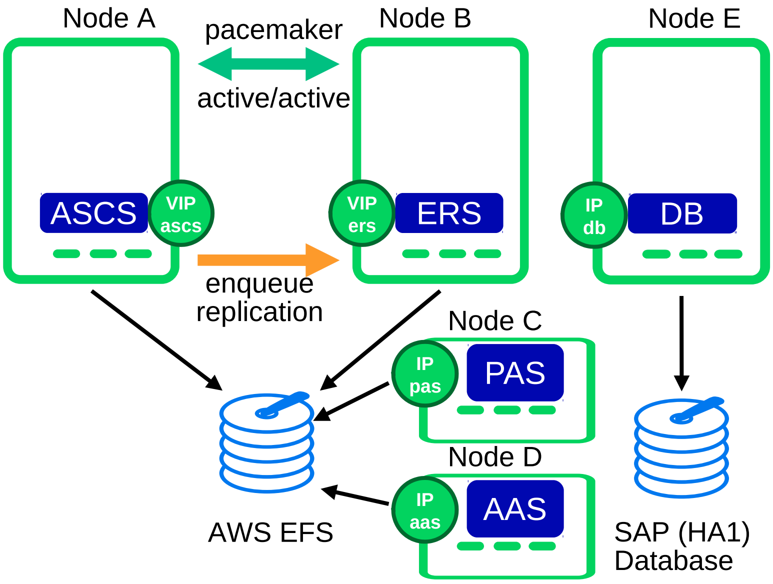

FIGURE 1: FIVE SYSTEMS FOR THE CERTIFICATION SETUP

CLUSTERED MACHINES

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

NON-CLUSTERED MACHINE

- one machine for DB; Hostname: sapha1db
- one machine for PAS; Hostname: sapha1ci
- one machine for AAS; Hostname: sapha1d2

3.7 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.

3.7.1 SAP HANA System Replication

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

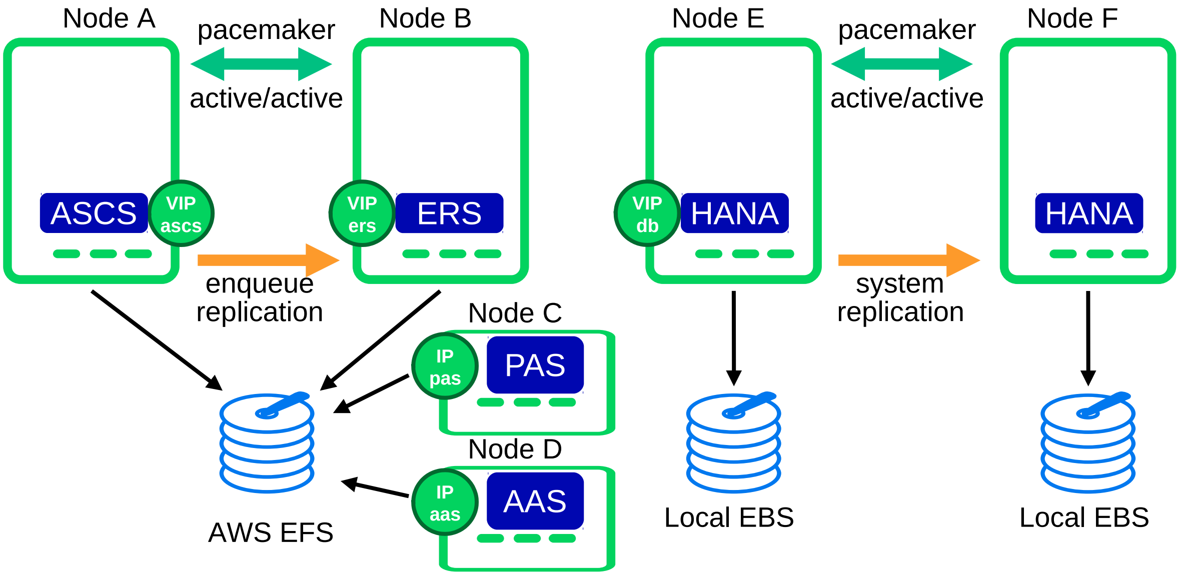

FIGURE 2: ONE CLUSTER FOR CENTRAL SERVICES, ONE FOR SAP HANA SR

The following Databases are supported in combination with this scenario:

- SAP HANA DATABASE 1.0
- SAP HANA DATABASE 2.0

3.8 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/ ↗).

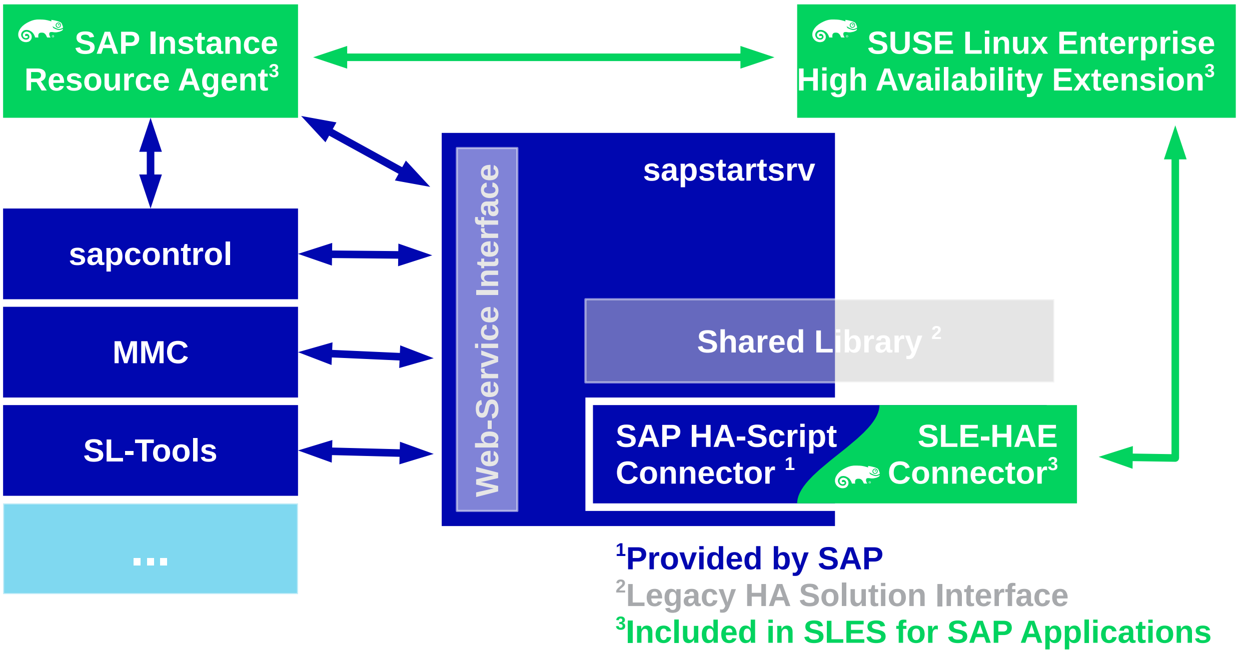
SAP Instance Resource Agent[3]
SUSE Linux Enterprise High Availability Extension[3]
sapstartsrv
Web-Service Interface
sapcontrol
MMC
SL-Tools
...
Shared Library [2]
SAP HA-Script Connector [1]
SLE-HAE Connector[3]
[1]Provided by SAP
[2]Legacy HA Solution Interface
[3]Included in SLES for SAP Applications

FIGURE 3: 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).

3.9 Disks and Partitions

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

3.9.1 EFS File Systems for Cluster ASCS and ERS

Create the following sub directories on both cluster nodes as root:

```
# mkdir -p /usr/sap/HA1/ASCS00 /usr/sap/HA1/ERS10
```

The file systems for the ASCS and ERS instances need to be shared and assigned to the cluster nodes hacert01 and hacert02. Create an EFS file system.

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

```
hacert01: efs-name:/ASCS00 /usr/sap/HA1/ASCS00
hacert02: efs-name:/ERS10 /usr/sap/HA1/ERS10
```

Replace the variable *efs-name* with the appropriate DNS name of the EFS filesystem.

Note

hacert01 and hacert02 operate in different availability zones. They need mount points named "efs-name" which are individual to the availability zone. Use the DNS name provided by AWS. The DNS name will point to the files system mount point local to a given Availability Zone. 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.

3.10 IP Addresses and Virtual Names

Check, if the file */etc/hosts* contains at least the following address resolutions. Add those entries, if they are missing. The 10.0.0.0 addresses in the example below are primary IP addresses within the VPC CIDR block. The 192.168.201.0 addresses are the Overlay IP addresses for the virtual services. The listing below lists a virtual IP address for the database server. A SAP system installation against a virtual database server address will allow to upgrade the database server to be a protected cluster service in a later step.

```
10.0.0.111  hacert01
10.0.0.112  hacert02
10.0.0.113  hacert03
10.0.0.114  sapha1ci
10.0.0.115  sapha1d2
192.168.201.116  sapha1as
192.168.201.117  sapha1er
192.168.201.118  sapha1db
```

3.11 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
- Replace *efs-name* with the appropriate DNS name.

EXAMPLE 1: SAP SAP NETWEAVER 7.4

```
# mkdir -p /sapmnt
# mkdir -p /usr/sap/HA1/{ASCS00,D02,DVEBMGS01,ERS10,SYS}
# mount -t nfs efs-name:/sapmnt /sapmnt
# mount -t nfs efs-name:/SYS    /usr/sap/HA1/SYS
# mount -t nfs efs-name:/sapcd /sapcd
```

EXAMPLE 2: SAP SAP NETWEAVER 7.5

```
# mkdir -p /sapmnt
# mkdir -p /usr/sap/HA1/{ASCS00,D01,D02,ERS10,SYS}
# mount -t nfs efs-name:/sapmnt    /sapmnt
# mount -t nfs efs-name:/SYS /usr/sap/HA1/SYS
# mount -t nfs efs-name:/sapcd /sapcd
```

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

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

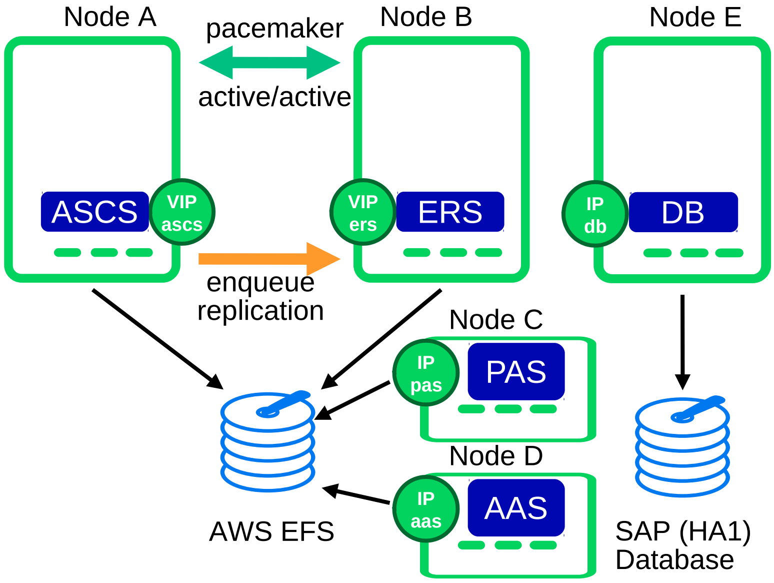

FIGURE 4: FILE SYSTEM LAYOUT INCLUDING NFS SHARES

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
- Server 3 (hacert03) will be used to install the database.
- Server 4 (sapha1ci) will be used to install the PAS SAP instance.

- Server 5 (sapha1d2) will be used to install the AAS SAP instance.
- Mounting the instance and database file systems at one specific node
- As a result the directory */usr/sap/HA1/* should now look like:

```
# ls -l /usr/sap/HA1/
total 0
drwxr-xr-x 1 ha1adm sapsys 70 28. Mär 17:26 ./
drwxr-xr-x 1 root   sapsys 58 28. Mär 16:49 ../
drwxr-xr-x 7 ha1adm sapsys 58 28. Mär 16:49 ASCS00/
drwxr-xr-x 1 ha1adm sapsys  0 28. Mär 15:59 D02/
drwxr-xr-x 1 ha1adm sapsys  0 28. Mär 15:59 DVEBMGS01/
drwxr-xr-x 1 ha1adm sapsys  0 28. Mär 15:59 ERS10/
drwxr-xr-x 5 ha1adm sapsys 87 28. Mär 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 Testing the AWS Agents

Test the AWS agents before you start up the cluster. The tests will show whether the AWS role and the policies are being configured correctly. The tests should excute without AWS CLI errors

4.1 Test of Overlay IP Agents

Replace the following variables in the commands

- *ip_address*: The service IP addresses of the ASCS and the ERS system
- *rtb-table* : The name of AWS routing table of the Overlay IP address
- *cluster* : replace the AWS CLI profile name if needed

The variables will have to match the variables in the OCF primitives later on!

Run the following commands as root on both systems:

```
OCF_RESKEY_address=ip_address \
OCF_RESKEY_routing_table=rtb-table \
OCF_RESKEY_interface=eth0 OCF_ROOT=/usr/lib/ocf OCF_RESKEY_profile=cluster \
/usr/lib/ocf/resource.d/suse/aws-vpc-move-ip start

OCF_RESKEY_address=ip_address \
OCF_RESKEY_routing_table=rtb-table \
OCF_RESKEY_interface=eth0 OCF_ROOT=/usr/lib/ocf OCF_RESKEY_profile=cluster \
/usr/lib/ocf/resource.d/suse/aws-vpc-move-ip monitor

OCF_RESKEY_address=ip_address \
OCF_RESKEY_routing_table=rtb-table \
OCF_RESKEY_interface=eth0 OCF_ROOT=/usr/lib/ocf OCF_RESKEY_profile=cluster \
/usr/lib/ocf/resource.d/suse/aws-vpc-move-ip stop
```

Check for AWS CLI access issues and fix the AWS Policy. Use the AWS console to check whether the IP address got added after *start*. Use the AWS console to check whether the IP got removed after *stop*. Use the other cluster node to execute some access commands (ping, ssh etc.) Recheck and fix all network related settings if this doesn’t work.

4.2 Test Agents mounting EFS File Sytems

Test the monitoring function first. Replace the following variable in the commands:

- *efs-name*: The name of the EFS filesystem

Run the following commands as root on both cluster nodes:

```
OCF_RESKEY_device="efs-name:ASCS00" \
OCF_RESKEY_directory="/usr/sap/HA1/ASCS00" OCF_RESKEY_fstype=nfs4 \
OCF_RESKEY_options="rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2" \
OCF_ROOT=/usr/lib/ocf /usr/lib/ocf/resource.d/heartbeat/Filesystem start

OCF_RESKEY_device="efs-name:ERS10" \
OCF_RESKEY_directory="/usr/sap/HA1/ERS10" OCF_RESKEY_fstype=nfs4 \
OCF_RESKEY_options="rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2" \
OCF_ROOT=/usr/lib/ocf /usr/lib/ocf/resource.d/heartbeat/Filesystem start

df -k

OCF_RESKEY_device="efs-name:ASCS00" \
OCF_RESKEY_directory="/usr/sap/HA1/ASCS00" OCF_RESKEY_fstype=nfs4 \
OCF_RESKEY_options="rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2" \
OCF_ROOT=/usr/lib/ocf /usr/lib/ocf/resource.d/heartbeat/Filesystem stop

OCF_RESKEY_device="efs-name:ERS10" \
OCF_RESKEY_directory="/usr/sap/HA1/ERS10" OCF_RESKEY_fstype=nfs4 \
OCF_RESKEY_options="rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2" \
OCF_ROOT=/usr/lib/ocf /usr/lib/ocf/resource.d/heartbeat/Filesystem stop

df -k
```

Check with the command *df -k* whether the filesystems got mounted and unmounted. Potential problems arise with an incorrect *efs-name* or with missing subdirectories.

4.3 Optional: Test of Route 53 Agents

This test should be conducted if the Route 53 agent will be used. Test the monitoring function first. Replace the following variables in the commands:

- *hosted zone id*: The ID of the hosted private zones
- *fullname* : The full name of the service name including sub domain and a trailing dot.
- *cluster* : replace the AWS CLI profile name if needed

The variables will have to match the variables in the OCF primitives later on!

Run the following commands as root on both systems:

```
OCF_RESKEY_hostedzoneid=hosted zone id OCF_RESKEY_ttl=10 \
    OCF_RESKEY_fullname=fullname OCF_ROOT=/usr/lib/ocf \
    OCF_RESKEY_profile=cluster \
    /usr/lib/ocf/resource.d/heartbeat/aws-vpc-route53 monitor

OCF_RESKEY_hostedzoneid=hosted zone id OCF_RESKEY_ttl=10 \
    OCF_RESKEY_fullname=fullname OCF_ROOT=/usr/lib/ocf \
    OCF_RESKEY_profile=cluster \
    /usr/lib/ocf/resource.d/heartbeat/aws-vpc-route53 start

OCF_RESKEY_hostedzoneid=hosted zone id OCF_RESKEY_ttl=10 \
  OCF_RESKEY_fullname=fullname OCF_ROOT=/usr/lib/ocf \
  OCF_RESKEY_profile=cluster \
  /usr/lib/ocf/resource.d/heartbeat/aws-vpc-route53 stop
```

Fix any problems in monitoring first. Try a start as second test and a stop as last test.

5 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:

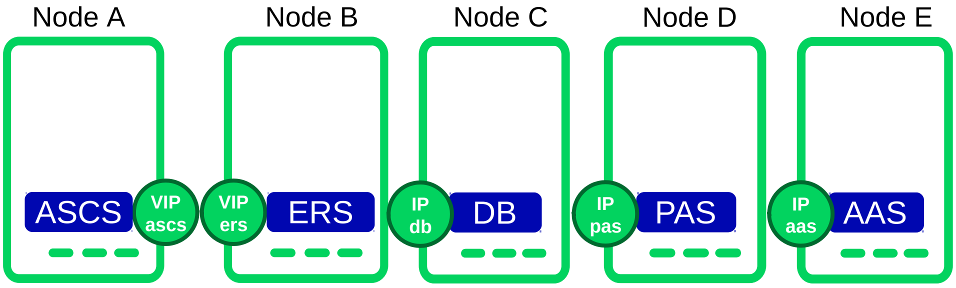

FIGURE 5: DISTRIBUTED INSTALLATION OF THE SAP SYSTEM

5.1 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  haladm       3000
User  sdb           3002
User  sqdha1        3003
User  sapadm        3004
User  h04adm        4001
```

5.2 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 efs-name:/ASCS00 and /sapcd/ which might also need to be mounted.

```
# ip address add 192.168.201.116/32 dev eth0
# mount efs-name:/ASCS00 /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

5.3 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 address add 192.168.201.117/32 dev eth0
# mount efs-name:/ERS10 /usr/sap/HA1/ERS10
# cd /sapcd/SWPM/
# ./sapinst SAPINST_USE_HOSTNAME=sapha1er
```

- SWPM option depends on SAP NetWeaver version and architecture
- Installing SAP NetWeaver 7.5 → 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 ha1adm:sapsys /usr/sap/HA1/ERS10
```

- If you get a prompt to manually stop/start the ASCS instance, please login at hacert01 as user ha1adm and call sapcontrol.

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

5.4 Poststeps for ASCS and ERS

5.4.1 Stopping ASCS and ERS

On hacert01

```
# su - ha1adm
# sapcontrol -nr 00 -function Stop
# sapcontrol -nr 00 -function StopService
```

On hacert02

```
# su - ha1adm
# sapcontrol -nr 10 -function Stop
# sapcontrol -nr 10 -function StopService
```

5.4.2 Maintaining *sapservices*

Ensure that the file */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
```

5.4.3 Integrating the Cluster Framework using the sap-suse-cluster-connector Package

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

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

Note

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 profile files HA1_ASCS00_sapha1as and HA1_ERS10_sapha1er in the profile directory */usr/sap/HA1/SYS/profile/*.

You need to tell the sapstartsrv service 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 ha1adm to the unix user group haclient.

```
# usermod -a -G haclient ha1adm
```

5.4.4 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)
```

5.4.5 Starting ASCS and ERS

On hacert01

```
# su - ha1adm
# sapcontrol -nr 00 -function StartService HA1
# sapcontrol -nr 00 -function Start
```

On hacert02

```
# su - ha1adm
# sapcontrol -nr 10 -function StartService HA1
# sapcontrol -nr 10 -function Start
```

5.5 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) and SAP HANA TDI (https://www.sap.com/documents/2015/03/74cdb554-5a7c-0010-82c7-eda71af511fa.html).

Install the HANA file systems as being described in the section "Planning the Deployment" of the AWS SAP HANA on the AWS Cloud: Quick Start Reference Deployment (http://docs.aws.amazon.com/quickstart/latest/sap-hana/planning.html)

Consider to install the database against an Overlay IP address which acts like a service IP address. This will allow to upgrade the database to run in a SLES for SAP HAE cluster:

- 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

5.6 Install the Primary Application Server (PAS) on sapha1ci

Add the following mount points to the /etc/fstab file on host sapha1ci. Replace the string "efs_fs_local_az" with the IP address of your EFS service in your availability zone.

```
efs-name:sapcd     /sapcd    nfs4    rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2
 0 0
efs-name:sapmnt    /sapmnt   nfs4    rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2
 0 0
efs-name:DVEBMGS01 /usr/sap/HA1/DVEBMGS01  nfs4
 rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2 0 0
efs-name:SYS       /usr/sap/HA1/SYS        nfs4
 rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2 0 0
```

Replace the variable *efs-name* with the appropriate DNS name.

Create mount directories and mount the file systems

```
# mkdir -p /sapcd /sapmnt /usr/sap/HA1/SYS /usr/sap/HA1/DVEBMGS01
# mount -a
```

Install the PAS server with the sapinst tool.

- 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

5.7 Install an Additional Application Server (AAS) on sapha1d2

Add the following mount points to the /etc/fstab file on host sapha1d2. Replace the string "efs_fs_local_az" with the IP address of your EFS service in your availability zone.

```
efs-name:sapcd     /sapcd           nfs4
 rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2 0 0
efs-name:sapmnt    /sapmnt          nfs4
 rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2 0 0
efs-name:D02       /usr/sap/HA1/D02  nfs4
 rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2 0 0
efs-name:SYS       /usr/sap/HA1/SYS  nfs4
 rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2 0 0
```

Replace the variable *efs-name* with the appropriate DNS name. Create mount directories and mount the file systems

```
# mkdir -p /sapcd /sapmnt /usr/sap/HA1/SYS /usr/sap/HA1/D02
# mount -a
```

Install the AAS server with the sapinst tool.

- 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 name **sapha1d2** is used

6 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

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.

TASKS

1. Setup NTP (best with yast2). Use AWS time service at 169.254.169.123 which is accessible from all EC2 instances. Enable ongoing synchronization.
2. Install pattern *ha_sles* on both cluster nodes

```
# zypper install -t pattern ha_sles
```

Activate the public cloud module to get updates for the AWS CLI (Command Line Interface):

```
# SUSEConnect --list-extensions
# SUSEConnect -p sle-module-public-cloud/12/x86_64
```

Update your packages with the command:

```
# zypper update
```

6.1 Configure the Cluster Base

TASKS

- Install and configure the cluster stack at the first machine

6.1.1 Configuration of System Logging

SUSE recommends to use rsyslogd for logging with the SUSE cluster. This is a default configuration. Some AWS AMIs however use syslogd logging. Please perform the following commands as root on all cluster nodes:

```
# zypper install rsyslog
```

Use option 1 (deinstallation of competing software, syslogd). Reboot both nodes.

6.1.2 Corosync Configuration

6.1.2.1 Configuration of the *corosync.conf* File

The configuration will have an IP address for node node-1 which is supposed to be ip-node-1. Node node-2 has an ip address to which we refer as ip-node-2.

All cluster nodes are required to have a local configuration file */etc/corosync/corosync.conf* which will be structured as follows.

The relevant information is being located in the two sections describing interface and nodelist. The other entries can be configured as needed for a specific implementation.

Note

AWS requires a specific manual corosync configuration.

Use the following configuration in the */etc/corosync/corosync.conf* file on both cluster nodes:

```
# Please read the corosync.conf.5 manual page
totem {
  version: 2
  token: 5000
  consensus: 7500
  token_retransmits_before_loss_const: 6
  crypto_cipher: none
  crypto_hash: none
  clear_node_high_bit: yes
  interface {
    ringnumber: 0
    bindnetaddr: <ip-local-node>
    mcastport: 5405
    ttl: 1
  }
  transport: udpu
}
 logging {
      fileline: off
      to_logfile: yes
      to_syslog: yes
      logfile: /var/log/cluster/corosync.log
      debug: off
      timestamp: on
      logger_subsys {
         subsys: QUORUM
         debug: off
     }
}
nodelist {
  node {
  ring0_addr: <ip-node-1>
  nodeid: 1
  }
  node {
  ring0_addr: <ip-node-2>
  nodeid: 2
  }
}

quorum {
  # Enable and configure quorum subsystem (default: off)
  # see also corosync.conf.5 and votequorum.5
  provider: corosync_votequorum
  expected_votes: 2
  two_node: 1
}
```

Replace the variables *ip-node-1* and *ip-node-2* with the IP addresses of your two cluster instances. Replace *ip-local-node* with the IP address of the server the file is being created.

The chosen settings for *crypto_cipher* and *crypto_hash* are suitable for clusters in AWS. They may be modified according to SUSE's documentation if strong encryption of cluster communication is desired.

6.1.3 Starting the Cluster

The next step is to start the cluster with the command on both nodes:

```
# systemctl start pacemaker
```

6.1.4 Checking the Configuration

The configuration can be checked with the command:

```
# corosync-cfgtool -s
```

It'll create a result like the following one for a cluster node with the IP address 10.0.0.111:

```
Printing ring status.
Local node ID 1
RING ID 0
id = 10.0.0.111
status = ring 0 active with no faults
```

The cluster in question has been using ring 0, the node had the ID 1.

- The *crm_mon -1* output should look like this:

```
Stack: corosync
Current DC: hacert01 (version 1.1.15-19.15-e174ec8) - partition with quorum
Last updated: Wed Dec  6 16:02:42 2017
Last change: Wed Dec  6 15:44:45 2017 by hacluster via crmd on hacert01

2 nodes configured
0 resources configured

Online: [ hacert01 hacert02 ]

Full list of resources:
```

6.2 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.

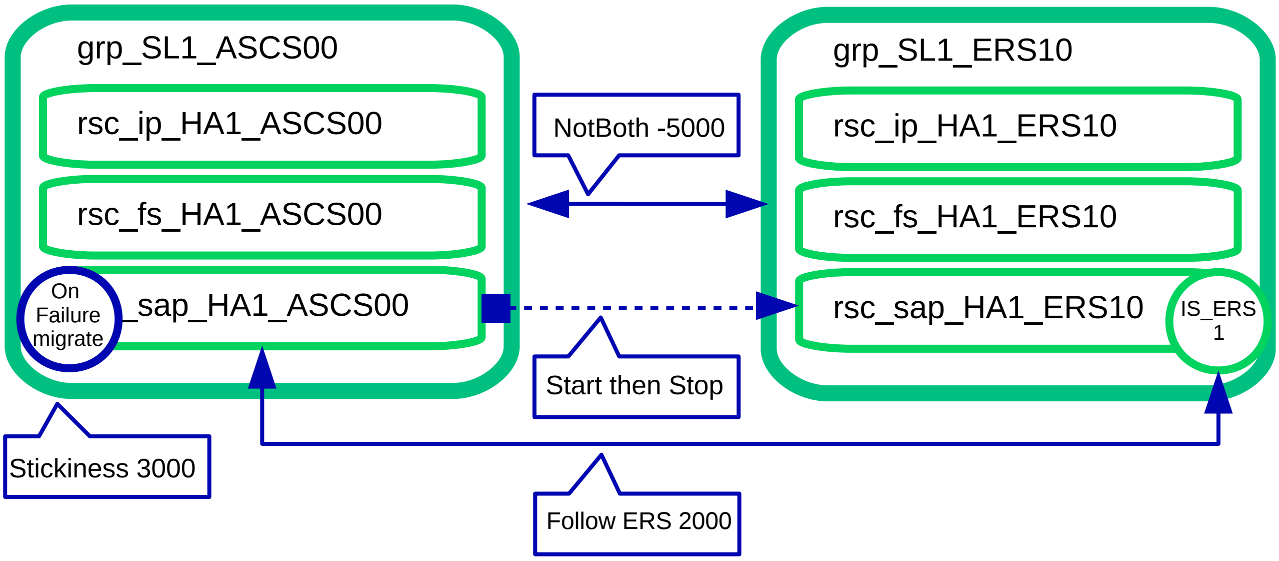

FIGURE 6: RESOURCES AND CONSTRAINTS

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".

There is the option to add a Route 53 agent. The architecture will then look as follows:

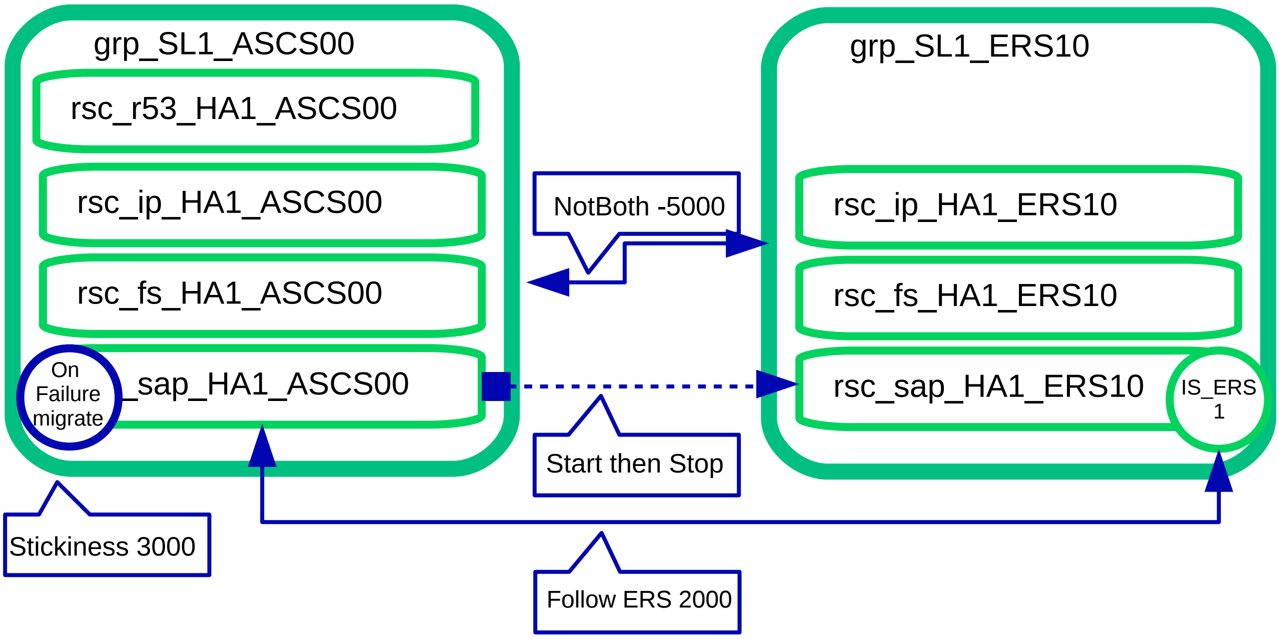

FIGURE 7: RESOURCES AND CONSTRAINTS

6.2.1 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"
```

6.2.2 Configure AWS specific Settings

Execute the following commands on one of the two cluster nodes:

```
# vi crm-bs.txt
```

Enter the following information to the file *crm-bs.txt*:

```
property cib-bootstrap-options: \
    stonith-enabled="true" \
    stonith-action="poweroff" \
    stonith-timeout="600s"
rsc_defaults rsc-options: \
        resource-stickiness=1 \
        migration-threshold=3
op_defaults op-options: \
        timeout=600 \
        record-pending=true
```

The setting *poweroff* forces the agents to shutdown the instance. This is desirable in order to avoid split brain scenarios on AWS.

Add the configuration to the cluster:

```
# crm configure load update crm-bs.txt
```

6.2.3 Configuration of AWS specific Stonith Resource

Create a file with the following content:

```
primitive res_AWS_STONITH stonith:external/ec2 \
op start interval=0 timeout=180 \
op stop interval=0 timeout=180 \
op monitor interval=120 timeout=60 \
params tag=pacemaker profile=cluster
```

The EC2 tag *pacemaker* entry needs to match the tag chosen for the EC2 instances. The value for this tag will contain the host name. The name of the profile (*cluster* in this example) will have to match the previously configured AWS profile.

Name this file for example *aws-stonith.txt* and add this file to the configuration. The following command has to be issued as root. It uses the file name *aws-stonith.txt*:

```
# crm configure load update aws-stonith.txt
```

6.2.4 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.

Create a file with your editor of choice with a name *aws-ascs.txt*. Add the ASCS primitive and the ASCS group to it. Don't forget to save your changes.

EXAMPLE 3: ASCS PRIMITIVE

```
primitive rsc_fs_HA1_ASCS00 Filesystem \
    params  device="efs-name:/ASCS00" directory="/usr/sap/HA1/ASCS00" \
            fstype="nfs4" \
            options="rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2" \
    op start timeout=60s interval=0 \
    op stop timeout=60s interval=0 \
    op monitor interval=200s timeout=40s
primitive rsc_ip_HA1_ASCS00 ocf:suse:aws-vpc-move-ip \
    params  address=192.168.201.116 routing_table=rtb-table \
            interface=eth0 profile=cluster \
    op start interval=0 timeout=180 \
    op stop interval=0 timeout=180 \
    op monitor interval=60 timeout=60
primitive rsc_sap_HA1_ASCS00 SAPInstance \
    operations $id=rsc_sap_HA1_ASCS00-operations \
    op monitor interval=120 timeout=60 on_fail=restart \
    params InstanceName=HA1_ASCS00_sapha1as \
        START_PROFILE="/sapmnt/HA1/profile/HA1_ASCS00_sapha1as" \
        AUTOMATIC_RECOVER=false \
    meta resource-stickiness=5000 failure-timeout=60 \
         migration-threshold=1 priority=10
```

Replace the variable *efs-name* with the name of your EFS server.

Replace the variable *rtb-table* with the identifier of the appropriate AWS routing table for the subnets. The name of the AWS CLI profile (*cluster* in this example) will have to match the previously configured AWS profile.

EXAMPLE 4: ASCS GROUP

```
group grp_HA1_ASCS00 \
  rsc_ip_HA1_ASCS00 rsc_fs_HA1_ASCS00 rsc_sap_HA1_ASCS00 \
        meta resource-stickiness=3000
```

Create a txt file *aws_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.

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

6.2.5 Optional: including Route 53

Name this file for example *aws-route53.txt* and add this file to the configuration. The following command has to be issued as root. It uses the file name *aws-route53.txt*:

Enter the following primitive before or after the existing primitives in the editor:

EXAMPLE 5: ROUTE53 PRIMITIVE

```
primitive rsc_r53_HA1_ASCS00 ocf:heartbeat:aws-vpc-route53 \
   params hostedzoneid=route-53-name ttl=10 fullname=name-full. profile=cluster \
   op start interval=0 timeout=180 \
   op stop interval=0 timeout=180 \
   op monitor interval=300 timeout=180
```

Replace the variable *route-53-name* with the name of the associated private hosted Route 53 zone.

Replace the variable *name-full.* will the fully qualified host name with matches the private hosted Route 53 zone.

The agent uses a time-to-live (ttl) of 10 seconds in this example. Change this parameter if needed.

Insert the *rsc_r53_HA1_ASCS00* after the *rsc_ip_HA1_ASCS00*. This will force the group to update then Route 53 as second item after the Overlay IP address.

EXAMPLE 6: ROUTE53 GROUP

```
group grp_HA1_ASCS00 \
  rsc_ip_HA1_ASCS00 rsc_r53_HA1_ASCS00 \
  rsc_fs_HA1_ASCS00 rsc_sap_HA1_ASCS00 \
        meta resource-stickiness=3000
```

Create a txt file *aws-route53.txt* with your preferred text editor, enter both examples (primitives and group) to that file and load the configuration to the cluster manager configuration. Use the following command as root and modify ASCS group in the editor.

```
# crm configure load update aws-route53.txt
```

Note

Version 1.0.2 of the Route 53 agent will not work if the EC2 metadata contains a string like "local-ipv4" in the userdata section!

6.2.6 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.

Replace *efs-name* with the name of your EFS server.

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="efs-name:/ERS10" directory="/usr/sap/HA1/ERS10" fstype=nfs4 \
  options="rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2" \
  op start timeout=60s interval=0 \
  op stop timeout=60s interval=0 \
  op monitor interval=200s timeout=40s
primitive rsc_ip_HA1_ERS10 ocf:suse:aws-vpc-move-ip \
  params address=192.168.201.117 routing_table=rtb-table \
  interface=eth0 profile=cluster \
  op start interval=0 timeout=180 \
  op stop interval=0 timeout=180 \
  op monitor interval=60 timeout=60
primitive rsc_sap_HA1_ERS10 SAPInstance \
  operations $id=rsc_sap_HA1_ERS10-operations \
  op monitor interval=120 timeout=60 on_fail=restart \
  params InstanceName=HA1_ERS10_sapha1er \
         START_PROFILE="/sapmnt/HA1/profile/HA1_ERS10_sapha1er" \
         AUTOMATIC_RECOVER=false IS_ERS=true \
  meta priority=1000
```

Replace the variable *rtb-table* with the identifier of the appropriate AWS routing table for the subnets. The name of the AWS CLI profile (*cluster* in this example) will have to match the previously configured AWS profile.

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 *aws_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 aws_crm_ers.txt
```

6.2.7 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 text 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.

Issue the following command as root:

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

6.2.8 Activating the Cluster

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

Issue the following command as root:

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

The cluster will now start the ASCS and the ERS system. This can can a few minutes. Check progress with the command:

```
# crm status
```

7 Administration

7.1 Do and Don’t Do

7.1.1 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.

7.1.2 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 ""
```

7.1.3 Never Block Resources

With SAP NW-HA-CLU 7.40 it is **not longer allowed to block resources** from beeing controlled manually. This using the variable *BLOCK_RESOURCES* in */etc/sysconfig/sap_suse_cluster_connector* is not allowed any more.

7.1.4 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.

7.1.5 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"
```

7.1.6 Procedure to End the Cluster Maintenance

As user root

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

7.1.7 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.

7.2 Testing the Cluster

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

7.2.1 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
```

7.2.2 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
```

7.2.3 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
```

7.2.4 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 ""
```

7.2.5 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
```

7.2.6 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
```

7.2.7 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

8 AWS specific Post Installation Tasks

The optional installation of the Route 53 agent will update the DNS name of the central instance as needed. The Route 53 naming tables may have to be made visible for on premises users like SAP GUI users. This happens through updating the on-premises name servers to delegate name resolution to Route 53. This forwarding of name resolution requests acquiresan extra configuration in the AWS VPC.

Active directory users will have to configure an Active Directory Connector as described in [https://aws.amazon.com/de/blogs/security/how-to-set-up-dns-resolution-between-on-premises-networks-and-aws-using-aws-directory-service-and-amazon-route-53/] .

DNS server users will have to implement bind forwarding EC2 instances as described in [http://www.scalingbits.com/aws/dnsfailover/backpropagation].

9 Appendix

9.1 CRM Config

The complete crm config for SAP system HA1

```
#
# nodes
#
node 1084753931: hacert01
node 1084753932: hacert02
#
# primitives for ASCS and ERS
#
primitive res_AWS_STONITH stonith:external/ec2 \
        op start interval=0 timeout=180 \
        op stop interval=0 timeout=180 \
        op monitor interval=120 timeout=60 \
        params tag=pacemaker profile=cluster
primitive rsc_fs_HA1_ASCS00 Filesystem \
  params device="efs-name:/ASCS00" \
         directory="/usr/sap/HA1/ASCS00" fstype=nfs4 \
         options="rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2" \
        op start timeout=60s interval=0 \
        op stop timeout=60s interval=0 \
        op monitor interval=200s timeout=40s
primitive rsc_fs_HA1_ERS10 Filesystem \
  params device="efs-name:/ERS10" \
    directory="/usr/sap/HA1/ERS10" fstype=nfs4 \
    options="rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2" \
        op start timeout=60s interval=0 \
        op stop timeout=60s interval=0 \
        op monitor interval=20s timeout=40s
primitive rsc_ip_HA1_ASCS00 ocf:suse:aws-vpc-move-ip \
        params address=192.168.201.116 routing_table=rtb-table-name \
        interface=eth0 profile=cluster \
        op start interval=0 timeout=180 \
        op stop interval=0 timeout=180 \
        op monitor interval=120 timeout=60
primitive rsc_ip_HA1_ERS10 ocf:suse:aws-vpc-move-ip \
        params address=192.168.201.117 routing_table=rtb-table-name \
        interface=eth0 profile=cluster \
        op start interval=0 timeout=180 \
        op stop interval=0 timeout=180 \
        op monitor interval=120 timeout=60
primitive rsc_r53_HA1_ASCS00 aws-vpc-route53 \
        params hostedzoneid=hosted-zone-id ttl=10 \
        fullname=full-name profile=cluster \
        op start interval=0 timeout=180 \
        op stop interval=0 timeout=180 \
        op monitor interval=300 timeout=180
primitive rsc_sap_HA1_ASCS00 SAPInstance \
        operations $id=rsc_sap_HA1_ASCS00-operations \
        op monitor interval=120 timeout=60 on_fail=restart \
        params InstanceName=HA1_ASCS00_sapha1as \
     START_PROFILE="/sapmnt/HA1/profile/HA1_ASCS00_sapha1as" \
     AUTOMATIC_RECOVER=false \
        meta resource-stickiness=5000 failure-timeout=60 migration-threshold=1 \
       priority=10
primitive rsc_sap_HA1_ERS10 SAPInstance \
        operations $id=rsc_sap_HA1_ERS10-operations \
        op monitor interval=120 timeout=60 on_fail=restart \
        params InstanceName=HA1_ERS10_sapha1er \
    START_PROFILE="/sapmnt/HA1/profile/HA1_ERS10_sapha1er" \
    AUTOMATIC_RECOVER=false IS_ERS=true \
        meta priority=1000
#
# group definitions for ASCS and ERS
#
group grp_HA1_ASCS00 rsc_ip_HA1_ASCS00 \
   rsc_r53_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=false \
    dc-version=1.1.15-21.1-e174ec8 \
    cluster-infrastructure=corosync \
    stonith-enabled=true \
    stonith-action=poweroff \
    stonith-timeout=600s \
    last-lrm-refresh=1513844735
rsc_defaults rsc-options: \
    resource-stickiness=1 \
    migration-threshold=3
op_defaults op-options: \
    timeout=600 \
        record-pending=true
```

9.2 Checklist AWS Installation

Check your AWS configuration upfront and gather the following AWS items before you start the installation:

Checklist AWS installation	
Item	**Status/Value**
SLES subscription and update status	
All systems have a SLES for SAP subscription	
All systems have a public cloud channel	
All system have been updated to use the latest patch level	
AWS User Privileges for the installing person	

Checklist AWS installation	
Creation of EC2 instances and EBS volumes	
Creation security groups	
Creation EFS file systems	
Modification of AWS routing tables	
Creation policies and attach them to IAM roles	
Optional for Route53 agent installation	
Create and modify A-records in a private hosted zone	
Potentially needed :Creation of subnets and routing tables	
VPC and Network	
VPC Id	
CIDR range of VPC	
Subnet id A for systems in first AZ	
Subnet id B for systems in second AZ	
Routing table id for subnet A and B	
Is this routing table associated with both subnets?	
Alternative: Is it associated to VPC? Subnets do not have their own ones	
Optional: Route 53 configuration	
Name of private hosted Route 53 zone	

Checklist AWS installation	
Name of DHCP option set (Verify options!)	
Is option set associated to VPC?	
AWS Policies Creation	
Name of data provider policy	
Name of STONITH policy	
Name of Move IP (Overlay IP) policy	
Optionally: Name of Route53 policy	
First cluster node (ASCS and ERS)	
instance id	
ENI id	
IP address	
hostname	
instance is associated to subnet A?	
instance has all 3 or 4 policies attached?	
EC2 tag *pacemaker* set with hostname?	
AWS CLI profile *cluster* created and set to *text*?	
source/destination check disabled?	
Second cluster node (ASCS and ERS)	
instance id	
ENI id	

Checklist AWS installation	
IP address	
hostname	
instance is associated to subnet B?	
instance has all 3 or 4 policies attached?	
EC2 tag *pacemaker* set with hostname?	
AWS CLI profile *cluster* created and set to *text*?	
source/destination check disabled?	
PAS system	
IP address	
hostname	
instance is associated to subnet A or B?	
instance has data provider policy attached?	
AAS system	
IP address	
hostname	
instance is associated to subnet A or B	
instance has data provider policy attached?	
DB system (is potentially node 1 of a database failover cluster)	
instance id	

Checklist AWS installation	
ENI id	
IP address	
hostname	
instance is associated to subnet A?	
instance has data provider policy attached? A cluster node has 2 to 3 more policies attached	
Overlay IP address: service ASCS	
IP address	
Has it been added to routing table?	
Does it point to the ENI of first node?	
Overlay IP address: service ERS	
IP address	
Has it been added to routing table?	
Does it point to the ENI of the second node?	
Optional: Overlay IP address DB server	
IP address	
Has it been added to routing table?	
Does it point to the ENI of the DB server?	
Optional: Route 53 configuration	
The Route 53 private hosted zone has an A record with the name of the	

Checklist AWS installation	
ASCS system the IP address of the first cluster node	
Creation of EFS filesystem	
DNS name of EFS filesystem	
Internet access	
All instance have Internet access ? Check routing tables	
Alternative: Add http proxies for data providers and cluster software	

9.3 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 ↗)