SAP HANA High Availability Cluster for the AWS Cloud

Setup Guide (v12)

SUSE Linux Enterprise Server for SAP Applications 12 SP5
Amazon Web Services

Fabian Herschel, Distinguished Architect SAP (SUSE)
Bernd Schubert, SAP Solution Architect (SUSE)
Lars Pinne, System Engineer (SUSE)
Guilherme Felix, Cloud Support Engineer (AWS)
Martin Tegtmeier, Principal Solutions Architect (AWS)
Stefan Schneider, Partner Solutions Architect (AWS)

SAP HANA High Availability Cluster for the AWS Cloud

Setup Guide (v12)

Date: 2024-12-09

SUSE® Linux Enterprise Server for SAP Applications is optimized in various ways for SAP* applications. This guide provides detailed information about installing and customizing SUSE Linux Enterprise Server for SAP Applications for SAP HANA system replication in the performance optimized scenario on the AWS platform. The document focuses on the steps to integrate an already installed and working SAP HANA with system replication. This document is based on SUSE Linux Enterprise Server for SAP Applications 12 SP5.

Contents

1 About This Guide

1.1 Introduction

SUSE® Linux Enterprise Server for SAP Applications is optimized in various ways for SAP*. This guide provides detailed information about installing and customizing SUSE Linux Enterprise Server for SAP Applications for SAP HANA system replication in the performance optimized scenario.

“SAP customers invest in SAP HANA” is the conclusion reached by a recent market study carried out by Pierre Audoin Consultants (PAC). In Germany, half of the companies expect SAP HANA to become the dominant database platform in the SAP environment. Often the “SAP Business Suite* powered by SAP HANA*” scenario is already being discussed in concrete terms.

SUSE is also accommodating this development by providing SUSE Linux Enterprise Server for SAP Applications – the recommended and supported operating system for SAP HANA. In close collaboration with SAP and hardware partners, SUSE provides two resource agents for customers to ensure the high availability of SAP HANA system replications.

1.1.1 Abstract

This guide describes planning, setup, and basic testing of SUSE Linux Enterprise Server for SAP Applications based on the high availability solution scenario "SAP HANA Scale-Up System Replication Performance Optimized".

From the application perspective the following variants are covered:

- Plain system replication
- System replication with secondary site read-enabled
- Multi-tier (chained) system replication
- Multi-target system replication
- Multi-tenant database containers for all above

From the infrastructure perspective the following variants are covered:

- 2-node cluster with AWS specific fencing

1.1.2 Scale-Up Versus Scale-Out

The first set of scenarios includes the architecture and development of scale-up solutions.

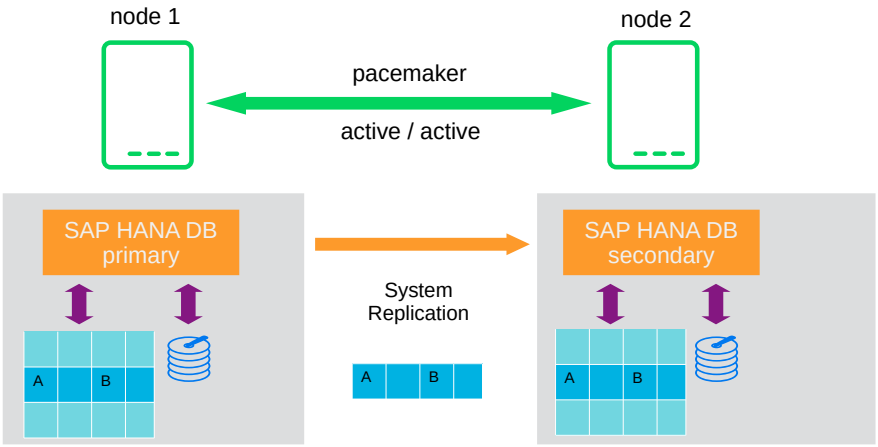


FIGURE 1: SAP HANA SYSTEM REPLICATION SCALE-UP IN THE CLUSTER

For this scenarios SUSE developed the scale-up resource agent package `SAPHanaSR`. System replication will help to replicate the database data from one computer to another to compensate for database failures (single-box replication).

The second set of scenarios includes the architecture and development of *scale-out* solutions (multi-box replication). For these scenarios SUSE developed the scale-out resource agent package `SAPHanaSR-ScaleOut`.

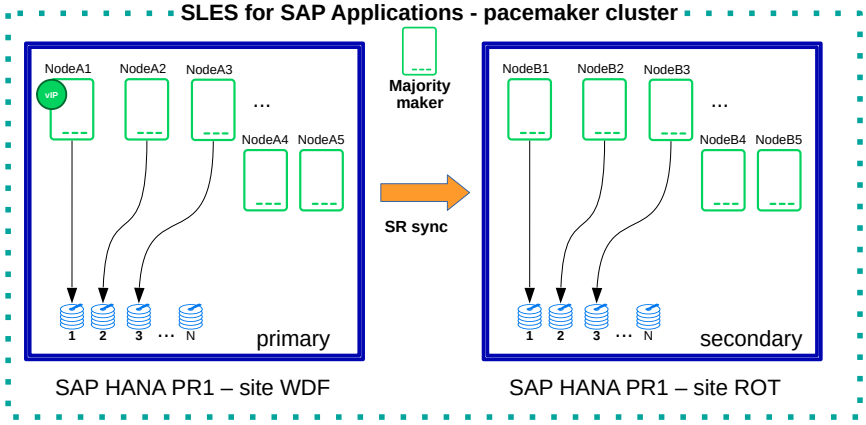


FIGURE 2: SAP HANA SYSTEM REPLICATION SCALE-OUT IN THE CLUSTER

With this mode of operation, internal SAP HANA high availability (HA) mechanisms and the resource agent must work together or be coordinated with each other. SAP HANA system replication automation for scale-out is described in a separate document available on our documentation Web page at https://documentation.suse.com/sbp/sap/. The document for scale-out is named *"SAP HANA System Replication Scale-Out High Availability in Amazon Web Services"*.

1.1.3 Scale-Up Scenarios and Resource Agents

SUSE has implemented the scale-up scenario with the `SAPHana` resource agent (RA), which performs the actual check of the SAP HANA database instances. This RA is configured as a master/slave resource. In the scale-up scenario, the master assumes responsibility for the SAP HANA databases running in primary mode. The slave is responsible for instances that are operated in synchronous (secondary) status.

To make configuring the cluster as simple as possible, SUSE has also developed the `SAPHanaTopology` resource agent. This RA runs on all nodes of a SUSE Linux Enterprise Server for SAP Applications cluster and gathers information about the statuses and configurations of SAP HANA system replications. It is designed as a normal (stateless) clone.

SAP HANA system replication for scale-up is supported in the following scenarios or use cases:

- **Performance optimized** ($A \Rightarrow B$). This scenario and setup **is described in this document.**

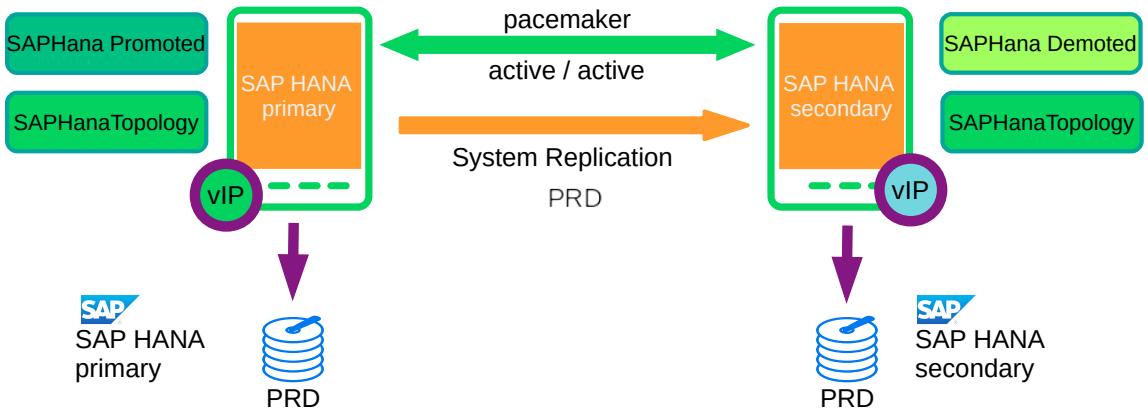


FIGURE 3: SAP HANA SYSTEM REPLICATION SCALE-UP IN THE CLUSTER - PERFORMANCE OPTIMIZED

In the performance optimized scenario, an SAP HANA RDBMS site A is synchronizing with an SAP HANA RDBMS site B on a second node. As the HANA RDBMS on the second node is configured to pre-load the tables, the takeover time is typically very short.

One big advance of the performance optimized scenario of SAP HANA is the possibility to allow read access on the secondary database site. To support this **read enabled** scenario, a second virtual IP address is added to the cluster and bound to the secondary role of the system replication.

- **Cost optimized** ($A \Rightarrow B, Q$). This scenario and setup is described in another document available from the documentation Web page at https://documentation.suse.com/sbp/sap/. The document for *cost optimized* is named *"Setting up a SAP HANA SR Cost Optimized Infrastructure"*.

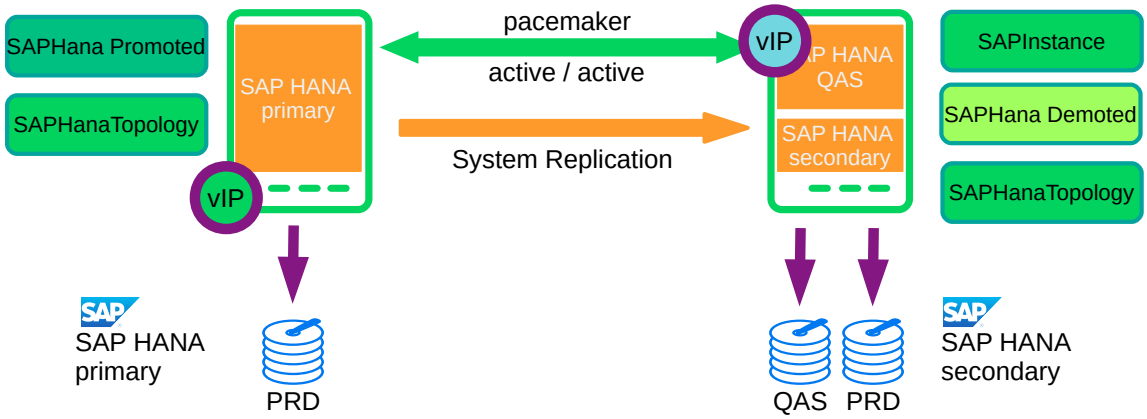


FIGURE 4: SAP HANA SYSTEM REPLICATION SCALE-UP IN THE CLUSTER - COST OPTIMIZED

In the cost optimized scenario, the second node is also used for a non-productive SAP HANA RDBMS system (like QAS or TST). Whenever a takeover is needed, the non-productive system must be stopped first. As the productive secondary system on this node must be limited in using system resources, the table preload must be switched off. A possible takeover needs longer than in the performance optimized use case.
In the cost optimized scenario, the secondary needs to be running in a reduced memory consumption configuration. This why *read enabled* must not be used in this scenario.

- **Multi Tier** ($A \Rightarrow B \rightarrow C$) and **Multi Target** ($B \Leftarrow A \Rightarrow C$).

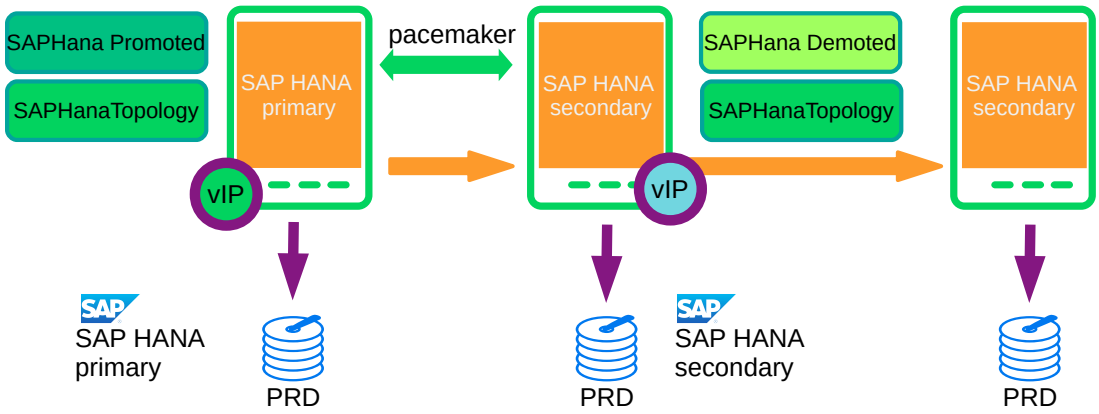


FIGURE 5: SAP HANA SYSTEM REPLICATION SCALE-UP IN THE CLUSTER - PERFORMANCE OPTIMIZED CHAIN

A *multi-tier* system replication has an additional target. In the past this third side must have been connected to the secondary (chain topology). With current SAP HANA versions, also *multiple target topology* is allowed by SAP.

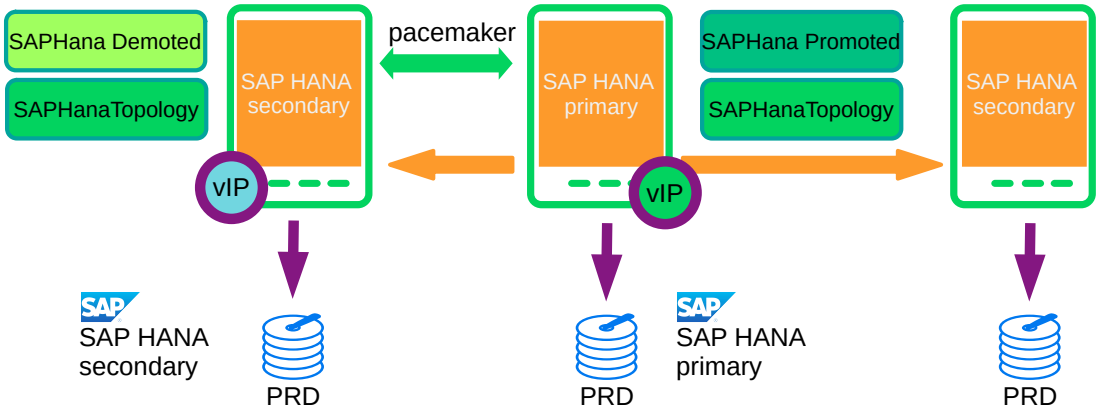


FIGURE 6: SAP HANA SYSTEM REPLICATION SCALE-UP IN THE CLUSTER - PERFORMANCE OPTIMIZED MULTI TARGET

Multi-tier and multi-target systems are implemented as described in this document. Only the first replication pair (A and B) is handled by the cluster itself. The main difference to the plain performance optimized scenario is that the auto registration must be switched off.

- **Multi-tenancy** or MDC.

Multi-tenancy is supported for all above scenarios and use cases. This scenario is supported since SAP HANA SPS09. The setup and configuration from a cluster point of view is the same for multi-tenancy and single containers. Thus you can use the above documents for both kinds of scenarios.

1.1.4 The Concept of the Performance Optimized Scenario

In case of failure of the primary SAP HANA on node 1 (node or database instance), the cluster first tries to start the takeover process. This allows to use the already loaded data at the secondary site. Typically the takeover is much faster than the local restart.

To achieve an automation of this resource handling process, use the SAP HANA resource agents included in SAPHanaSR. System replication of the productive database is automated with SAPHana and SAPHanaTopology.

The cluster only allows a takeover to the secondary site if the SAP HANA system replication was in sync until the point when the service of the primary got lost. This ensures that the last commits processed on the primary site are already available at the secondary site.

SAP did improve the interfaces between SAP HANA and external software such as cluster frameworks. These improvements also include the implementation of SAP HANA call outs in case of special events such as status changes for services or system replication channels. These call outs are also called HA/DR providers. This interface can be used by implementing SAP HANA hooks written in python. SUSE improved the SAPHanaSR package to include such SAP HANA hooks to optimize the cluster interface. Using the SAP HANA hook described in this document allows to inform the cluster immediately if the SAP HANA system replication breaks. In addition to the SAP HANA hook status, the cluster continues to poll the system replication status on a regular basis.

You can set up the level of automation by setting the parameter `AUTOMATED_REGISTER`. If automated registration is activated, the cluster will also automatically register a former failed primary to get the new secondary.

Important

The solution is not designed to manually 'migrate' the primary or secondary instance using HAWK or any other cluster client commands. In the *Administration* section of this document we describe how to 'migrate' the primary to the secondary site using SAP and cluster commands.

1.1.5 Customers Receive Complete Package

Using the SAPHana and SAPHanaTopology resource agents, customers can integrate SAP HANA system replications in their cluster. This has the advantage of enabling companies to use not only their business-critical SAP systems but also their SAP HANA databases without interruption while noticeably reducing needed budgets. SUSE provides the extended solution together with best practices documentation.

SAP and hardware partners who do not have their own SAP HANA high availability solution will also benefit from this development from SUSE.

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 find numerous white papers, best practices, setup guides, and other resources on the SUSE Linux Enterprise Server for SAP Applications best practices Web page at https://documentation.suse.com/sbp/sap/.

SUSE also publishes blog articles about SAP and high availability using the hashtag #TowardsZeroDowntime. For more information, follow the link https://www.suse.com/c/tag/TowardsZeroDowntime/.

1.3 Errata

To deliver urgent smaller fixes and important information in a timely manner, the Technical Information Document (TID) for this setup guide will be updated, maintained and published at a higher frequency:

- SAP HANA SR Performance Optimized Scenario - Setup Guide - Errata (https://www.suse.com/support/kb/doc/?id=7023882)
- Showing SOK Status in Cluster Monitoring Tools Workaround (https://www.suse.com/support/kb/doc/?id=7023526 - see also the blog article https://www.suse.com/c/lets-flip-the-flags-is-my-sap-hana-database-in-sync-or-not/)

In addition to this guide, check the SUSE SAP Best Practice Guide Errata for other solutions (https://www.suse.com/support/kb/doc/?id=7023713).

1.4 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).

Mail

For feedback on the documentation of this product, you can send a mail to doc-team@suse.com (mailto:doc-team@suse.com). 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 Supported Scenarios and Prerequisites

With the `SAPHanaSR` resource agent software package, we limit the support to scale-up (single-box to single-box) system replication with the following configurations and parameters:

- Two-node cluster.
- The cluster must include a valid STONITH method.
- The AWS EC2 STONITH mechanism supported by SUSE Linux Enterprise High Availability Extension 12 is supported with SAPHanaSR.
- Each cluster node is in a different Availability Zone (AZ) within the same AWS Region.
- The Overlay IP address must be an IP outside the Virtual Private Cloud (VPC) CIDR.
- Technical users and groups, such as *<sid>adm*, are defined locally in the Linux system.
- Name resolution of the cluster nodes and the virtual IP address must be done locally on all cluster nodes.
- Time synchronization between the cluster nodes like NTP is required.
- Both SAP HANA instances (primary and secondary) have the same SAP Identifier (SID) and instance number.

- If the cluster nodes are installed in different AWS Availability Zones, the environment must match the requirements of the SLE HAE cluster product. Of particular concern is the network latency and recommended maximum distance between the nodes. Review the product documentation for SUSE Linux Enterprise High Availability Extension regarding those recommendations.
- Automated registration of a failed primary after takeover is available.
- SAP HANA Replication mode should be set to **SYNC** or **SYNCMEM** - **ASYNC** is not supported by the cluster.
- SAP HANA Replication operation mode can be either *logreplay*, *logreplay_readaccess* or *delta_datashipping*.
    - As a good starting configuration for projects, we recommend to switch off the automated registration of a failed primary. The setup `AUTOMATED_REGISTER="false"` is the default. In this case, you need to register a failed primary after a takeover manually. Use SAP tools like SAP HANA cockpit or *hdbnsutil*.
    - For optimal automation, we recommend `AUTOMATED_REGISTER="true"`.
- Automated start of SAP HANA instances during system boot must be switched off.
- Multi-tenancy (MDC) databases are supported.
    - Multi-tenancy databases could be used in combination with any other setup (performance based, cost optimized and multi-tier).
    - In MDC configurations the SAP HANA RDBMS is treated as a single system including all database containers. Therefore, cluster takeover decisions are based on the complete RDBMS status independent of the status of individual database containers.
    - For SAP HANA 1.0 you need version SPS10 rev3, SPS11 or newer if you want to stop tenants during production and if you want the cluster to be able to take over. Older SAP HANA versions are marking the system replication as failed if you stop a tenant.
    - Tests on multi-tenancy databases could force a different test procedure if you are using strong separation of the tenants. As an example, killing the complete SAP HANA instance using *HDB kill* does not work, because the tenants are running with different Linux user UIDs. <sidadm> is not allowed to terminate the processes of the other tenant users.

You need at least SAPHanaSR version 0.152 and in best SUSE Linux Enterprise Server for SAP Applications 12 SP4 or newer. SAP HANA 1.0 is supported since SPS09 (095) for all mentioned setups. SAP HANA 2.0 is supported with all known SPS versions.

Important

Without a valid STONITH method, the complete cluster is unsupported and will not work properly.

If you need to implement a different scenario, we strongly recommend to define a Proof of Concept (PoC) with SUSE. This PoC will focus on testing the existing solution in your scenario. Most of the above mentioned limitations exist because careful testing is needed.

Besides SAP HANA, you need SAP Host Agent to be installed on your system.

3 Scope of This Document

This document describes how to set up the cluster to control SAP HANA in System Replication Scenarios. The document focuses on the steps to integrate an already installed and working SAP HANA with System Replication.

The described example setup builds an SAP HANA HA cluster in two Availability Zones in one AWS Region. Availability Zone 1 is "A" and Availability Zone 2 is "B", installed on two SUSE Linux Enterprise Server for SAP Applications 12 SP5 systems.

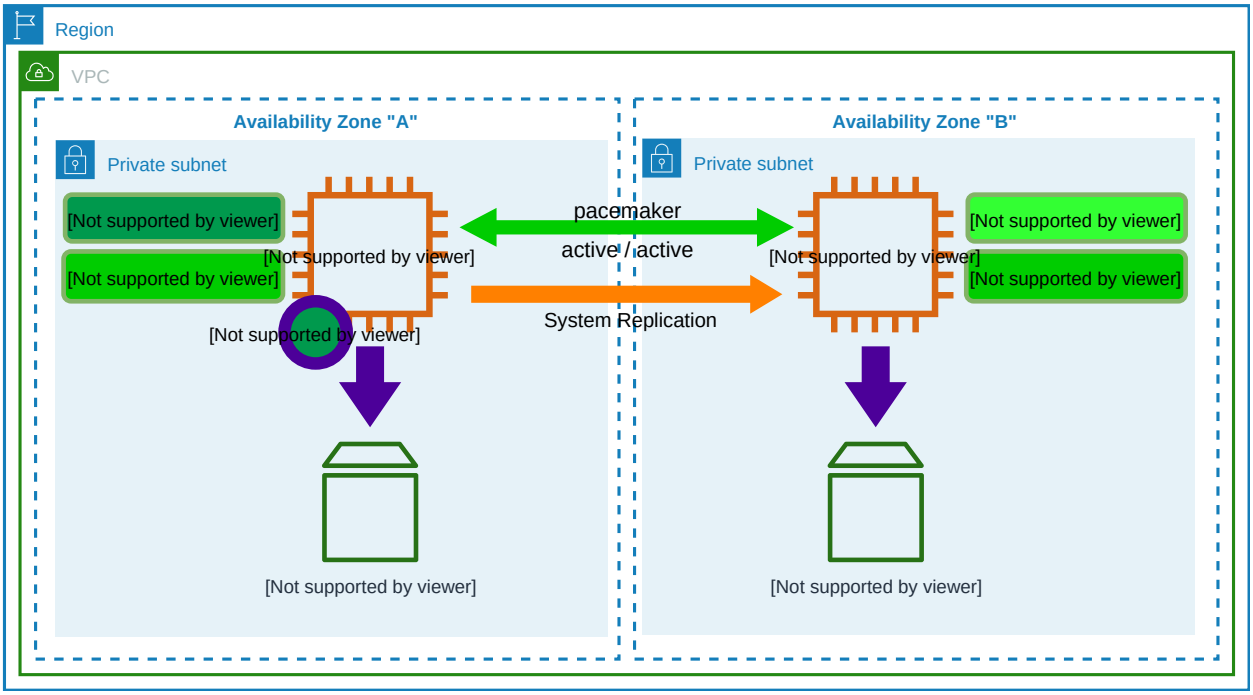
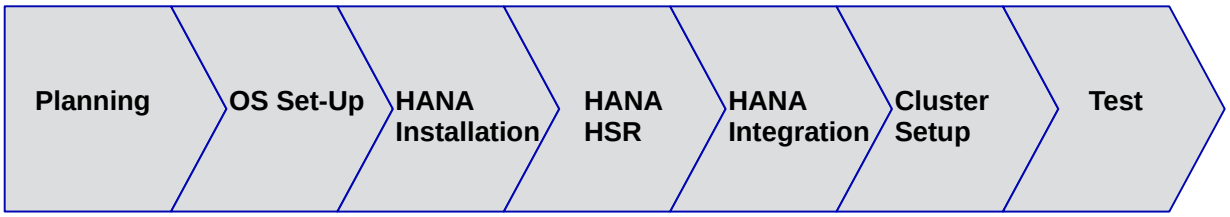


FIGURE 7: CLUSTER WITH SAP HANA SR - PERFORMANCE OPTIMIZED

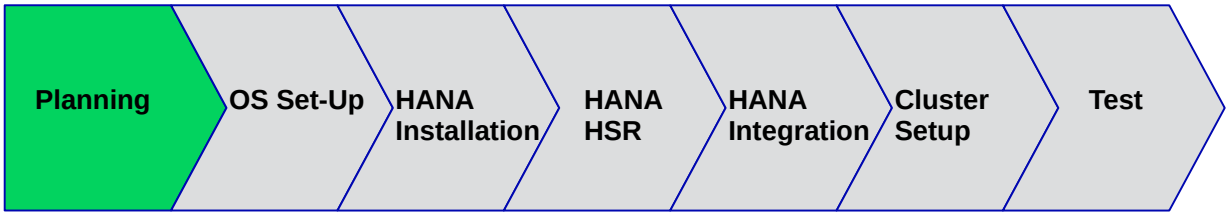
This guide focuses on the manual setup of the cluster to explain the details and to give you the possibility to create your own automation.

The seven main setup steps are:



- Planning (see *Section 4, "Planning the Installation"*)
- Operating system installation (see *Section 5, "Setting up the Operating System"*)
- Database installation (see *Section 6, "Installing the SAP HANA Databases on Both Cluster Nodes"*)
- SAP HANA system replication setup (see *Section 7, "Setting Up SAP HANA System Replication"*
- SAP HANA HA/DR provider hooks (see *Section 8, "Setting Up SAP HANA HA/DR Providers"*)
- Cluster configuration (see *Section 9, "Configuring the Cluster"*)
- Testing (see *Section 10, "Testing the Cluster"*)

4 Planning the Installation



Planning the installation is essential for a successful SAP HANA cluster setup.

What you need before you start:

- Understand your AWS infrastructure and architecture
- (Optional) Software from SUSE: a valid SUSE subscription, and access to update channels
- Software from SAP: SAP HANA installation media
- Two AWS EC2 instances in different Availability Zones
- Filled parameter sheet (see below)

TABLE 1: PARAMETERS USED IN THIS DOCUMENT

Parameter	Value	Role
Cluster node 1	suse01, 192.168.1.11,192.168.1.12	Cluster node name and IP addresses.
Cluster node 2	suse02, 192.168.2.11,192.168.2.12	Cluster node name and IP addresses.
SID	HA1	SAP Identifier
Instance number	10	Number of the SAP HANA database. For system replication also Instance Number + 1 is blocked.
Network mask	255.255.255.0	
Virtual IP address	10.0.0.1	
Storage		Storage for HDB data and log files is connected “locally” (per node; not shared)

Note

The preferred method to deploy SAP HANA Scale-Up clusters in AWS is to use the AWS Launch Wizard for SAP (https://docs.aws.amazon.com/launchwizard/latest/user-guide/launch-wizard-sap.html). However, if you are installing SAP HANA Scale-Up manually, refer to the AWS SAP HANA Guides (https://docs.aws.amazon.com/sap/latest/sap-hana/welcome.html) for detailed installation instructions, including recommended storage configuration and file systems.

4.1 AWS Requirements for SUSE Linux Enterprise Server Clusters

SUSE Linux Enterprise Server pacemaker clusters will run in an AWS region.

An AWS region consists of multiple independent Availability Zones (AZs), which is one or more discrete data centers with redundant power, networking, and connectivity in an AWS Region. AZs give customers the ability to operate production applications and databases that are more highly available, fault tolerant, and scalable than would be possible from a single data center. All AZs in an AWS Region are interconnected with high-bandwidth, low-latency networking, over fully redundant, dedicated metro fiber providing high-throughput, low-latency networking between AZs. All traffic between AZs is encrypted. The network performance is sufficient to accomplish synchronous replication between AZs.

An AWS Virtual Private Network (VPC) spans all AZs within an AWS Region, thus the following is required:

- Select two Availability Zones within an AWS Region for the SAP HANA cluster implementation.
- Identify one subnet in each AZ to host the cluster nodes.
- Use one or more VPC routing tables which are attached to the two subnets being used.
- Optionally, host a Route53 private hosted naming zone to manage names in the VPC.
- All components of the cluster and AWS services should reside in the same AWS account. The use of networking components such as a VPC route table in another account (Shared VPC setup) is not supported. If a multi account landscape is required, we advise you reach to your AWS representative to have a look at implementing a Transit Gateway for cross account/VPC access.

The virtual IP address for the SAP HANA will be an AWS Overlay IP address. This is an AWS specific routing table entry which will send network traffic to an instance, no matter which AZ the instance is located in. The SUSE Linux Enterprise High Availability Extension cluster updates this VPC routing table entry as needed.

The Overlay IP addresses needs to be different from the VPC CIDR range. All SAP system components within the VPC can reach an AWS EC2 instance through this Overlay IP address.

On-premises users and clients, like SAP HANA Studio, cannot reach the Overlay IP address because the AWS Virtual Private Network (VPN) gateway is not able to route traffic to the Overlay IP address. To overcome this limitation, refer to AWS' Overlay IP documentation and learn how to use native AWS services with the Overlay IP address for your on-premises clients and users:

- SAP on AWS High Availability with Overlay IP Address Routing: https://docs.aws.amazon.com/sap/latest/sap-hana/sap-ha-overlay-ip.html

Below are the prerequisites which need to be met before starting the cluster implementation:

- Have an AWS account
- Have an AWS user with admin privileges, or with permissions to:
  - Create or modify VPC Security Groups
  - Modify AWS VPC Routing Tables
  - Create IAM policies and attach them to IAM roles
  - Create and Modify EC2 Instances
- Understand your architecture:
  - Know your AWS Region and its AWS name
  - Know your VPC and its AWS VPC ID
  - Know which Availability Zones you want to use in your VPC
  - Have the VPC Subnet for each of the AZs:
    - Have one or more routing tables which are implicitly or explicitly attached to the two subnets
    - Have free IP addresses in the two VPC Subnets

- Allow network traffic in between the two subnets
- Allow outgoing Internet access from the subnets

Use the checklist in the appendix to note down all information needed before starting the installation.

4.2 Security Groups

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

- Port 5405 for inbound UDP: Required by the cluster's communication layer (corosync).
- Port 7630 for inbound TCP: Used by the SUSE "HAWK" Web GUI.

It is assumed that there are no restrictions for outbound network communication.

4.3 Creating AWS EC2 Instance

Create two EC2 instances to build up your SUSE Linux Enterprise High Availability Extension cluster.

The EC2 instances must be located in two different Availability Zones to make them independent of each other, and it is recommended to be one of the certified SAP HANA instances as per the SAP HANA Certified Hardware Directory (https://www.sap.com/dmc/exp/2014-09-02-hana-hardware/enEN/#/solutions).

There are two options for which Amazon Machine Image (AMI) to use:

- Use the AWS Marketplace AMI *"SUSE Linux Enterprise Server for SAP Applications 12 SP5"* which already includes the required SUSE subscription and all High Availability components for this solution.
- Use a "SUSE Linux Enterprise Server for SAP" AMI. Search for *"suse-sles-sap-12-sp5-byos"* in the list of AMIs. There are several BYOS (Bring Your Own Subscription) AMIs available. Use these AMIs if you have a valid SUSE subscription. Register your system with the Subscription Management Tool (SMT) from SUSE, SUSE Manager or directly with the SUSE Customer Center.

Launch all EC2 instances into the Availability Zones (AZ) specific subnets. The subnets need to be able to communicate with each other.

Note

It is not possible to migrate from standard *"SUSE Linux Enterprise Server"* to *"SUSE Linux Enterprise Server for SAP Applications"* in AWS. Therefore, use a "SLES for SAP" AMI which includes the SUSE Linux Enterprise High Availability Extension.

4.4 Tagging the EC2 Instances

The AWS EC2 STONITH agents use AWS resource tags to identify the EC2 instances.

Tag the two EC2 instances through the console or the AWS Command Line Interface (CLI) with arbitrarily chosen tags like *pacemaker* and the host name as it will be shown in the command *uname*. Use the same tag (like *pacemaker*) and the individual host names for both instances.

To add a tag to an EC2 instance, refer to the AWS Documentation: * Tagging your Amazon EC2 resources: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Using_Tags.html

See an example screenshot after the EC2 instance has been tagged. A tag with the key pacemaker and the host name has been created. The host name in this example is *suse-node52*.

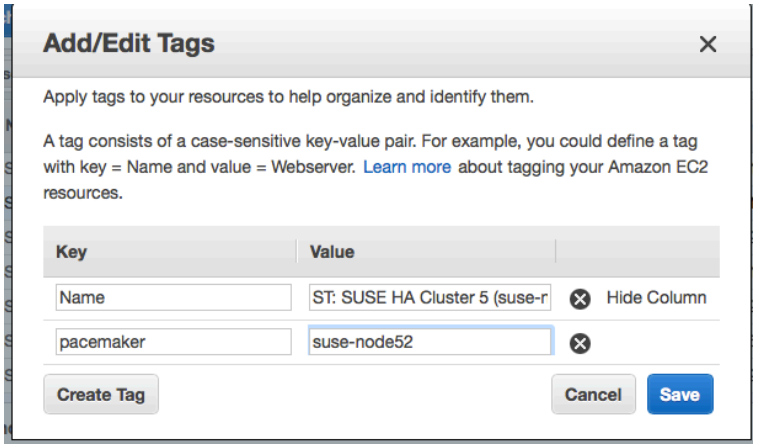


FIGURE 8: TAG EC2 INSTANCE

Make sure that both EC2 instances part of the cluster are tagged.

Note

Use only ASCII characters in any AWS tag assigned to cluster managed resources.

4.4.1 Disabling Source/Destination Check for Cluster Instances

The source/destination check needs to be disabled. This can be done through scripts using the AWS CLI or by using the AWS console.

The following command needs to be executed one time for both EC2 instances that are part of the cluster:

EXAMPLE 1: DISABLING SOURCE/DESTINATION CHECK USING AWS CLI

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

Replace the variable *EC2-instance* with the EC2 instance IDs of the two cluster AWS EC2 instances.

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

EXAMPLE 2: IAM POLICY REQUIRED TO CHANGE SOURCE/DESTINATION CHECK

```
{
   "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 following individual parameters with the appropriate values:

- region-name : the name of the AWS region
- account-id : The number of the AWS account in which the policy is used
- instance-a and instance-b : The two EC2 instance ids participating in the cluster

The source/destination check can be also disabled from the AWS console. It requires the following action in the console on both EC2 instances (see below).

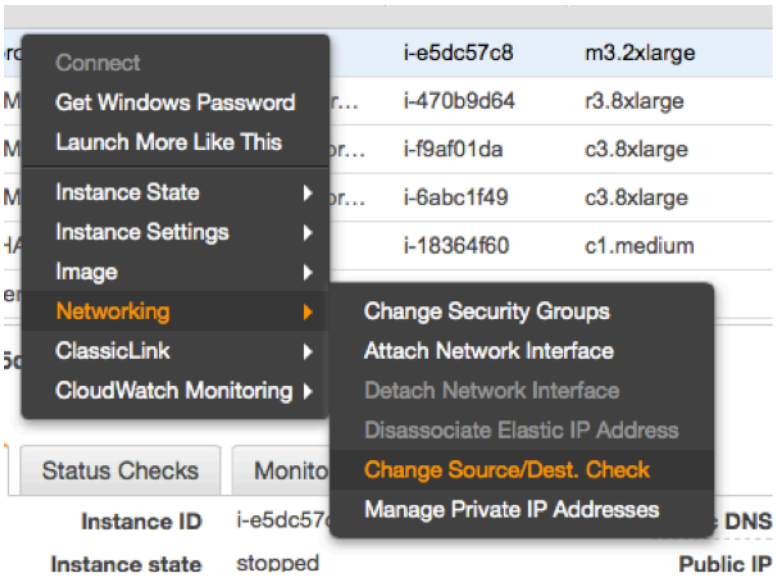


FIGURE 9: DISABLE SOURCE/DESTINATION CHECK AT CONSOLE

4.5 AWS Roles and Policies Required by the Cluster

The SAP HANA database EC2 instances will run the SUSE Linux Enterprise Server cluster software and its agents. To operate the cluster correctly, it requires specific AWS IAM privileges.

Create a new IAM Role for every SAP HANA *cluster* and associate this IAM Role to the two EC2 instances part of the cluster. Attach the following IAM Policies to this IAM Role.

4.5.1 AWS Data Provider Policy

Every cluster node will operate an SAP system. SAP systems on AWS require the installation of the *"AWS Data Provider for SAP"*. The data provider needs a policy to pull information from AWS resources.

The policy shown below can be used by all SAP systems as the *"AWS Data Provider for SAP"* can have only one policy per AWS account. Therefore you can use an existing one, previously created for the *"AWS Data Provider for SAP"*, or create a new one.

The *"AWS Data Provider for SAP"* IAM policy does not contain any EC2 instance specific privileges. Attach this IAM policy to the IAM role of the two cluster instances.

EXAMPLE 3: IAM POLICY FOR AWS DATA PROVIDER FOR SAP

```
{
    "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-sap-data-provider/config.properties"
        }
    ]
}
```

For more details about the permissions required by the AWS Data Provider for SAP, refer to AWS public documentation: * AWS Data Provider for SAP: https://docs.aws.amazon.com/sap/latest/general/aws-data-provider.html

4.5.1.1 EC2 STONITH IAM Permissions

The EC2 instances part of the cluster must have permission to make start and stop API calls to the other nodes in the cluster as part of the fencing operation. Create an IAM policy with a name like *EC2-stonith-policy* with the following content and attach it to the cluster IAM Role:

EXAMPLE 4: IAM POLICY FOR EC2 STONITH

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

```
        {
            "Sid": "Stmt1424870324001",
            "Effect": "Allow",
            "Action": [
                "ec2:RebootInstances",
                "ec2:StartInstances",
                "ec2:StopInstances"
            ],
            "Resource": [
                "arn:aws:ec2:region-name:account-id:instance/instance-a",
                "arn:aws:ec2:region-name:account-id:instance/instance-b"

            ]
        }
    ]
}
```

This policy allows the EC2 STONITH agent to make the proper API calls to operate correctly. From the above example, replace the following variables with the appropriate names:

- region-name : The name of the AWS region
- account-id : The number of the AWS account in which the policy is used
- instance-a and instance-b : The two EC2 instance IDs participating in the cluster

4.5.2 Overlay IP Resource Agent IAM Policy

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

EXAMPLE 5: IAM POLICY FOR AWS IP RESOURCE AGENT

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "ec2:ReplaceRoute",
            "Resource": "arn:aws:ec2:region-name:account-id:route-table/rtb-XYZ"
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
```

```
            "Action": "ec2:DescribeRouteTables",
            "Resource": "*"
        }
    ]
}
```

This policy allows the agent to update the routing table(s) where the Overlay IP address has been configured. From the above example, replace the following variables with the appropriate names:

- region-name : The name of the AWS region
- account-id : The number of the AWS account in which the policy is used
- rtb-XYZ : The VPC routing table identifier to be configured by the cluster. It is possible to add more routing table IDs to the resource clause if you need to use multiple routing tables.

4.6 Adding Overlay IP Addresses to Routing Tables

Manually add the Overlay IP address as a routing entry to the VPC routing tables which are assigned to the subnets. The Overlay IP address is the virtual service IP address of the SAP HANA cluster. The Overlay IP address needs to be outside of the CIDR range of the VPC.

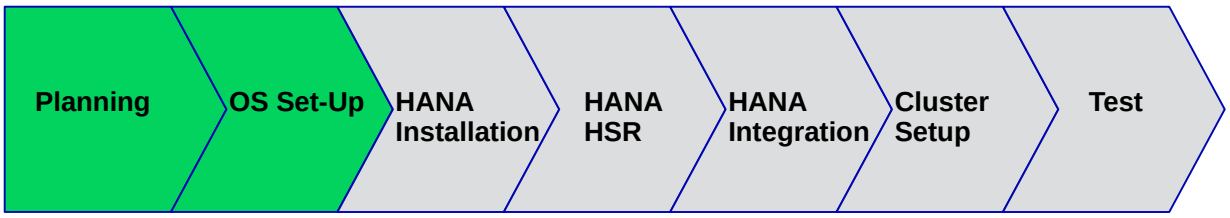
To add the Overlay IP address, do the following:

- Use the AWS console and search for “VPC”.
- Select the correct VPC ID.
- Click “Route Tables” in the left column.
- Select the route table used by the subnets from one of your SAP EC2 instances and their application servers.
- Click the tabulator “Routes”.
- Click “Edit”.
- Scroll to the end of the list and click “Add another route”.
- Add the Overlay IP address of the SAP HANA database. Use as filter /32 (example: 192.168.10.1/32). Add the Elastic Network Interface (ENI) name to one of your existing instance. The resource agent will modify this later automatically.
- Save your changes by clicking “Save”.

Note

The VPC routing table containing the routing entry needs to be inherited to all subnets in the VPC which have consumers or clients of the service. Add more routing tables if required. Check the AWS VPC documentation at http://docs.aws.amazon.com/Amazon-VPC/latest/UserGuide/VPC_Introduction.html for more details on routing table inheritance.

5 Setting up the Operating System



This section contains information you should consider during the installation of the operating system.

For the scope of this document, first SUSE Linux Enterprise Server for SAP Applications is configured. Then the SAP HANA database including the system replication is set up. Finally the automation with the cluster is set up and configured.

5.1 Setting System Host Name

The EC2 instances will have host names which are automatically generated, and these automatically generated host names must be changed. Select host names which comply with SAP requirements, see SAP Note 611361.

To change the host name you need to edit */etc/cloud/cloud.cfg* and change the option *preserve_hostname* to *true* for host names to persist:

EXAMPLE 6: OPTION CHANGED IN CLOUD.CFG FILE

```
preserve_hostname: true
```

Note

To learn how to change the default host name for an EC2 instance running SUSE Linux Enterprise, refer to the AWS' public documentation at https://aws.amazon.com/premium-support/knowledge-center/linux-static-hostname-suse/.

5.1.1 Configuring System Logging

SUSE recommends to use `rsyslogd` for logging in the SUSE cluster. Despite of this being the default configuration on newer AMIs, some AWS AMIs may still be using `syslogd` logging.

Perform the following commands as *root* on all cluster nodes:

EXAMPLE 7: SUSE LINUX ENTERPRISE SERVER 12 RSYSLOG INSTALLATION

```
suse01:~> zypper install rsyslog
```

Depending on the installed packages, a conflict may be shown, like in the below example:

```
suse01:~ # zypper install rsyslog
Refreshing service 'SMT-http_smt-ec2_susecloud_net'.
Refreshing service 'cloud_update'.
Loading repository data...
Reading installed packages...
Resolving package dependencies...
Problem: syslog-ng-3.6.4-11.1.x86_64 conflicts with namespace:otherproviders(syslog)
provided by rsyslog-8.24.0-3.16.1.x86_64
 Solution 1: deinstallation of syslog-ng-3.6.4-11.1.x86_64
 Solution 2: do not install rsyslog-8.24.0-3.16.1.x86_64
Choose from above solutions by number or cancel [1/2/c] (c):
```

Select "Solution 1: deinstallation of syslog-ng", and then reboot both nodes.

Additionally, some cluster components require `ha_logd` to properly log events, thus it needs to be set to start at boot:

EXAMPLE 8: ENABLING LOGD TO START AUTOMATICALLY

```
suse01:~> systemctl enable --now logd
```

5.2 Configuring the AWS CLI in the EC2 Instances

The SUSE Linux Enterprise Server agents use the AWS Command Line Interface (CLI) as an underlying tool to make AWS API calls.

It will use an AWS CLI profile which needs to be created for the user *root* on both instances. The SUSE resources agents require a profile that creates output in text format.

The name of the AWS CLI 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:

EXAMPLE 9: AWS CLI CONFIGURATION FILE

```
[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:

EXAMPLE 10: AWS CLI PROFILE CREATION

```
# 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.

5.3 Configuring 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 points are usually directly reachable. Systems which do not offer transparent Internet access need to provide an HTTP/HTTPS proxy. The configuration of the proxy access is described in full detail in the AWS documentation.

Add the following environment variables to the root user's *.bashrc* and to */etc/sysconfig/pacemaker* files:

EXAMPLE 11: ENVIRONMENT VARIABLES FOR PROXY

```
export HTTP_PROXY=http://a.b.c.d:n
export HTTPS_PROXY=http://a.b.c.d:m
export NO_PROXY=169.254.169.254
```

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

EXAMPLE 12: ENVIRONMENT VARIABLES FOR PROXY WITH AUTHENTICATION

```
export HTTP_PROXY=http://username:password@a.b.c.d:n
export HTTPS_PROXY=http://username:password@a.b.c.d:m
export NO_PROXY=169.254.169.254
```

There is also the option to configure the proxy system wide, which is detailed in the following SUSE Support Knowledgebase article:

- SUSE Linux Enterprise : How to set up a Proxy manually (https://www.suse.com/support/kb/doc/?id=000017441)

5.3.1 Verifying HTTP Proxy Settings

Make sure that the EC2 instance can communicate with the EC2 metadata server URL at http://169.254.169.254/latest/meta-data .

An incorrect configuration will cause issues to the SUSE registration and to the EC2 STONITH agent.

5.4 Configuring the Operating System for SAP HANA

The main installation guides for SUSE Linux Enterprise Server for SAP Applications that fit all requirements for SAP HANA are available from the following SAP notes:

- 1984787 SUSE LINUX Enterprise Server 12: Installation notes
- 2205917 SAP HANA DB: Recommended OS settings for SLES 12 / SLES for SAP Applications 12.

Other related SAP Notes are the following: - 1275776 Linux: Preparing SLES for SAP environments - 2382421 Optimizing the Network Configuration on HANA- and OS-Level

5.5 Managing Networking for Cluster Instances

5.5.1 Adding a Second IP for Each Cluster Instance

The cluster configuration requires two IP addresses per cluster instance, as corosync requires a redundant communication ring.

The redundant corosync ring configuration allows the cluster nodes to communicate with each other using the secondary IP address if there is an issue communicating with each other over the primary IP address. This avoids unnecessary cluster failovers and split-brain situations.

Refer to the AWS documentation at https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/MultipleIP.html#assignIP-existing to understand how to assign a secondary IP address.

After the secondary IP address is associated to the cluster instance in AWS, you need to configure the secondary IP address in the cluster instance. Update the file */etc/sysconfig/network/ifcfg-eth0* as shown below. Replace XX.XX.XX.XX with the new secondary IP address and replace 'XX' with the two digit subnet mask.

EXAMPLE 13: SECONDARY IP ADDRESS CONFIGURATION

```
IPADDR_1="XX.XX.XX.XX/XX"
LABEL_1="1"
```

The system will read the file and add the secondary IP address after the cluster instance is rebooted. Additionally, executing the command below as *root* will add the IP address to the cluster instance network stack without rebooting.

EXAMPLE 14: SECONDARY IP ADDRESS CONFIGURATION

```
ip address add XX.XX.XX.XX/XX dev eth0
```

Replace *XX.XX.XX.XX* with the new secondary IP address and replace *XX* with the two digit subnet mask.

5.5.2 Avoiding Deletion of Cluster Managed IP Address from the Network Interface

SUSE Linux Enterprise Server ships with the `cloud-netconfig-ec2` package which contains scripts to automatically configure network interfaces in an EC2 instance.

This package may remove secondary IP addresses which are managed by the cluster agents from the network interface. This can cause service interruptions for users of the cluster services. Perform the following task on all cluster nodes:

Check whether the package `cloud-netconfig-ec2` is installed with the command.

EXAMPLE 15: CHECK IF CLOUD-NETCONFIG-EC2 IS INSTALLED

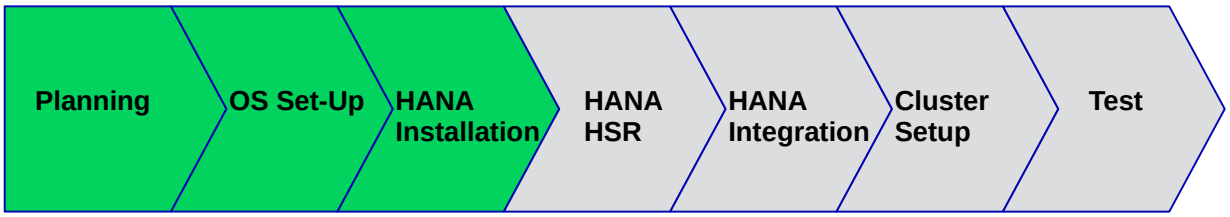
```
# zypper info cloud-netconfig-ec2
```

If this package is installed, update the file */etc/sysconfig/network/ifcfg-eth0* and change the following line to a **no** setting. If the package is not yet installed, add the following line:

EXAMPLE 16: DISABLING CLOUD_NETCONFIG_MANAGE

```
CLOUD_NETCONFIG_MANAGE='no'
```

6 Installing the SAP HANA Databases on Both Cluster Nodes



Even though this document focuses on the integration of an installed SAP HANA with system replication already set up into the pacemaker cluster, this chapter summarizes the test environment. Always use the official documentation from SAP to install SAP HANA and to set up the system replication.

PREPARATION

- Read the SAP Installation and Setup Manuals available at the SAP Marketplace.
- Download the SAP HANA Software from the SAP Marketplace.

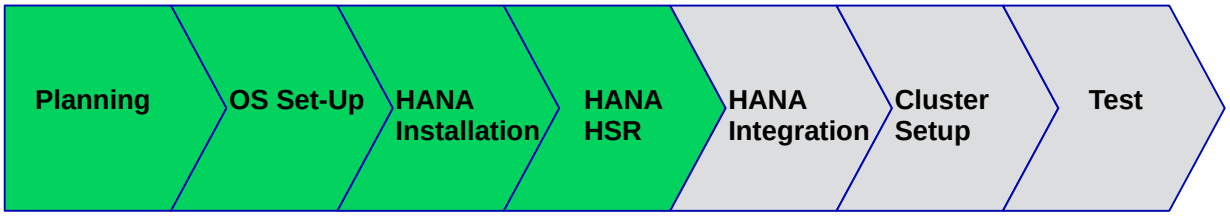
ACTIONS

1. Install the SAP HANA Database as described in the SAP HANA Server Installation Guide.
2. Check if the SAP Host Agent is installed on all cluster nodes. If this SAP service is not installed, install it now.
3. Verify that both databases are up and all processes of these databases are running correctly.

As Linux user *<sid>adm*, use the command line tool `HDB` to get an overview of the running HANA processes. The output of `HDB info` should be similar to the output shown below:

```
suse02:~> HDB info
USER          PID  ...  COMMAND
ha1adm        6561 ...  -csh
ha1adm        6635 ...    \_ /bin/sh /usr/sap/HA1/HDB10/HDB info
ha1adm        6658 ...        \_ ps fx -U ha1 -o user,pid,ppid,pcpu,vsz,rss,args
ha1adm        5442 ...  sapstart pf=/hana/shared/HA1/profile/HA1_HDB10_suse02
ha1adm        5456 ...   \_ /usr/sap/HA1/HDB10/suse02/trace/hdb.sapHA1_HDB10 -d
-nw -f /usr/sap/HA1/HDB10/suse
ha1adm        5482 ...       \_ hdbnameserver
ha1adm        5551 ...       \_ hdbpreprocessor
ha1adm        5554 ...       \_ hdbcompileserver
ha1adm        5583 ...       \_ hdbindexserver
ha1adm        5586 ...       \_ hdbstatisticsserver
ha1adm        5589 ...       \_ hdbxsengine
ha1adm        5944 ...       \_ sapwebdisp_hdb
pf=/usr/sap/HA1/HDB10/suse02/wdisp/sapwebdisp.pfl -f /usr/sap/SL
ha1adm        5363 ...  /usr/sap/HA1/HDB10/exe/sapstartsrv
pf=/hana/shared/HA1/profile/HA1_HDB10_suse02 -D -u s
```

7 Setting Up SAP HANA System Replication



For more information, read the section *Setting Up System Replication* of the SAP HANA Administration Guide.

Procedure

1. Back up the primary database.
2. Enable the primary database.
3. Register the secondary database.
4. Verify the system replication.

7.1 Backing Up the Primary Database

Back up the primary database as described in the SAP HANA Administration Guide, section *SAP HANA Database Backup and Recovery*. We provide an example with SQL commands. You need to adapt these backup commands to match your backup infrastructure.

EXAMPLE 17: SIMPLE BACKUP FOR THE SYSTEM DATABASE AND ALL TENANTS WITH ONE SINGLE BACKUP CALL

As user <*sidadm*> enter the following command:

```
hdbsql -u SYSTEM -d SYSTEMDB \
   "BACKUP DATA FOR FULL SYSTEM USING FILE ('backup')"
```

You will get the following command output (or similar):

```
0 rows affected (overall time 15.352069 sec; server time 15.347745 sec)
```

EXAMPLE 18: SIMPLE BACKUP FOR A SINGLE CONTAINER (NON MDC) DATABASE

Enter the following command as user <*sidadm*>:

```
hdbsql -i <instanceNumber> -u <dbuser> \
   "BACKUP DATA USING FILE ('backup')"
```

Important

Without a valid backup, you cannot bring SAP HANA into a system replication configuration.

7.2 Enabling the Primary Node

As Linux user <*sid*>*adm*, enable the system replication at the primary node. You need to define a site name (like WDF). This site name must be unique for all SAP HANA databases which are connected via system replication. This means the secondary must have a different site name.

Note

Do not use strings like "primary" and "secondary" as site names.

EXAMPLE 19: ENABLE THE PRIMARY

Enable the primary using the `-sr_enable` option.

```
suse01:~> hdbnsutil -sr_enable --name=WDF
checking local nameserver:
checking for active nameserver ...
nameserver is running, proceeding ...
configuring ini files ...
successfully enabled system as primary site ...
done.
```

EXAMPLE 20: CHECK SR CONFIGURATION ON THE PRIMARY

Check the primary using the command `hdbnsutil -sr_stateConfiguration`.

```
suse01:~> hdbnsutil -sr_stateConfiguration --sapcontrol=1
SAPCONTROL-OK: <begin>
mode=primary
site id=1
site name=WDF
SAPCONTROL-OK: <end>
done.
```

The mode has changed from “none” to “primary” and the site now has a site name and a site ID.

7.3 Registering the Secondary Node

The SAP HANA database instance on the secondary side must be stopped before the instance can be registered for the system replication. You can use your preferred method to stop the instance (like `HDB` or `sapcontrol`). After the database instance has been stopped successfully, you can register the instance using `hdbnsutil`. Again, use the Linux user *<sid>adm*:

EXAMPLE 21: STOP THE SECONDARY

To stop the secondary you can use the command line tool *HDB*.

```
suse02:~> HDB stop
```

EXAMPLE 22: COPY THE KEY AND KEY-DATA FILE FROM THE PRIMARY TO THE SECONDARY SITE

Beginning with SAP HANA 2.0, the system replication is running encrypted. This is why the key files need to be copied over from the primary to the secondary site.

```
cd /usr/sap/<SID>/SYS/global/security/rsecssfs
rsync -va {,<node1-siteB>:}$PWD/data/SSFS_<SID>.DAT
rsync -va {,<node1-siteB>:}$PWD/key/SSFS_<SID>.KEY
```

EXAMPLE 23: REGISTER THE SECONDARY

The registration of the secondary is triggered by calling `hdbnsutil -sr_register …`.

```
...
suse02:~> hdbnsutil -sr_register --name=ROT \
     --remoteHost=suse01 --remoteInstance=10 \
     --replicationMode=sync --operationMode=logreplay
adding site ...
checking for inactive nameserver ...
nameserver suse02:30001 not responding.
collecting information ...
updating local ini files ...
done.
```

The *remoteHost* is the primary node in our case, the *remoteInstance* is the database instance number (here 10).

Now start the database instance again and verify the system replication status. On the secondary node, the mode should be one of "SYNC" or "SYNCMEM". "ASYNC" is also a possible replication mode **but not supported with automated cluster takeover**. The mode depends on the "sync" option defined during the registration of the secondary.

EXAMPLE 24: START SECONDARY AND CHECK SR CONFIGURATION

To start the new secondary, use the command line tool `HDB`. Then check the SR configuration using `hdbnsutil -sr_stateConfiguration`.

```
suse02:~> HDB start
...
suse02:~> hdbnsutil -sr_stateConfiguration --sapcontrol=1
SAPCONTROL-OK: <begin>
mode=sync
site id=2
site name=ROT
active primary site=1
primary masters=suse01
SAPCONTROL-OK: <end>
done.
```

To view the replication state of the whole SAP HANA cluster, use the following command as *<sid>adm* user on the primary node:

EXAMPLE 25: CHECKING SYSTEM REPLICATION STATUS DETAILS

> The python script *systemReplicationStatus.py* provides details about the current system replication.
>
> ```
> suse01:~> HDBSettings.sh systemReplicationStatus.py --sapcontrol=1
> ...
> site/2/SITE_NAME=ROT1
> site/2/SOURCE_SITE_ID=1
> site/2/REPLICATION_MODE=SYNC
> site/2/REPLICATION_STATUS=ACTIVE
> site/1/REPLICATION_MODE=PRIMARY
> site/1/SITE_NAME=WDF1
> local_site_id=1
> ...
> ```

7.4 Manually Testing SAP HANA SR Takeover

Before you integrate your SAP HANA system replication into the cluster, it is mandatory to do a manual takeover. Testing without the cluster helps to make sure that basic operation (takeover and registration) is working as expected.

- Stop SAP HANA on node 1
- Takeover SAP HANA to node 2
- Register node 1 as secondary
- Start SAP HANA on node 1
- Wait until sync state is active

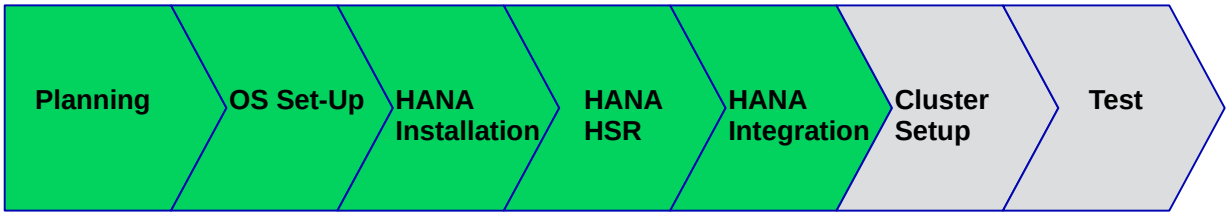
7.5 Optional: Manually Re-Establishing SAP HANA SR to Original State

Bring the systems back to the original state:

- Stop SAP HANA on node 2
- Takeover SAP HANA to node 1

- Register node 2 as secondary
- Start SAP HANA on node2
- Wait until sync state is active

8 Setting Up SAP HANA HA/DR Providers



This step is mandatory to inform the cluster immediately if the secondary gets out of sync. The hook is called by SAP HANA using the HA/DR provider interface at that point of time when the secondary gets out of sync. This is typically the case when the first commit pending is released. The hook is called by SAP HANA again when the system replication is back.

Procedure

1. Implement the python hook SAPHanaSR.
2. Configure the system replication operation mode.
3. Allow <*sidadm*> to access the cluster.
4. Start SAP HANA.
5. Test the hook integration.

8.1 Implementing the Python Hook SAPHanaSR

This step must be done on both sites. SAP HANA must be stopped to change the *global.ini* file and allow SAP HANA to integrate the HA/DR hook script during start.

- Install the HA/DR hook script into a read/writable directory.
- Integrate the hook into the *global.ini* file (SAP HANA needs to be stopped for doing that offline).
- Check the integration of the hook during start-up.

Use the hook from the SAPHanaSR package (available since version 0.153). Optionally copy it to your preferred directory like */hana/share/myHooks*. The hook must be available on all SAP HANA cluster nodes.

EXAMPLE 26: STOP SAP HANA

> Stop SAP HANA either with `HDB` or using `sapcontrol`.
>
> ```
> sapcontrol -nr <instanceNumber> -function StopSystem
> ```

EXAMPLE 27: ADDING SAPHANASR VIA GLOBAL.INI

```
[ha_dr_provider_SAPHanaSR]
provider = SAPHanaSR
path = /usr/share/SAPHanaSR
execution_order = 1

[trace]
ha_dr_saphanasr = info
```

8.2 Configuring System Replication Operation Mode

When your system is connected as an SAPHanaSR target you can find an entry in the *global.ini* file which defines the operation mode. Up to now there are the following modes available:

- *delta_datashipping*
- *logreplay*
- *logreplay_readaccess*

Until a takeover and re-registration in the opposite direction, the entry for the operation mode is missing on your primary site. The first operation mode which was available was *delta_datashipping*. Today the preferred modes for HA are *logreplay* or *logreplay_readaccess*. Using the operation mode *logreplay* makes your secondary site in the SAP HANA system replication a hot standby system. For more details regarding all operation modes check the available SAP documentation such as "How To Perform System Replication for SAP HANA".

EXAMPLE 28: CHECKING THE OPERATION MODE

> Check both *global.ini* files and add the operation mode if needed.

section

[system_replication]

entry

operation_mode = logreplay

Path for the *global.ini*: /hana/shared/<SID>/global/hdb/custom/config/

```
[system_replication]
operation_mode = logreplay
```

8.3 Allowing <sidadm> to Access the Cluster

The current version of the SAPHanaSR python hook uses the command `sudo` to allow the *<sidadm>* user to access the cluster attributes. In Linux you can use `visudo` to start the vi editor for the */etc/sudoers* configuration file.

The user *<sidadm>* must be able to set the cluster attributes `hana_<sid>_site_srHook_*`. The SAP HANA system replication hook needs password free access. The following example limits the sudo access to exactly setting the needed attribute.

Replace the <sid> by the **lowercase** SAP system ID (like `ha1`).

EXAMPLE 29: ENTRY IN SUDO PERMISSIONS /ETC/SUDOERS FILE

Basic sudoers entry to allow <sidadm> to use the `srHook`:

```
# SAPHanaSR-ScaleUp entries for writing srHook cluster attribute
<sidadm> ALL=(ALL) NOPASSWD: /usr/sbin/crm_attribute -n hana_<sid>_site_srHook_*
```

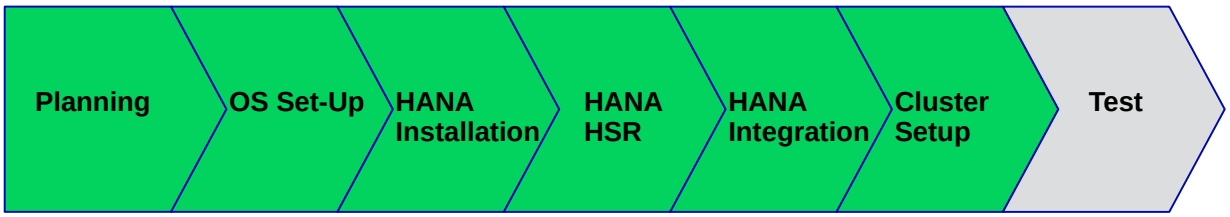
More specific sudoers entries to meet a high security level:

All **Cmnd_Alias** entries must be each defined as a single line entry. In the following example the lines might include a line break forced by document formatting. In our example we have four separate lines with Cmnd_Alias entries, one line for the *<sidadm>* user and one or more lines for comments.

```
# SAPHanaSR-ScaleUp entries for writing srHook cluster attribute
Cmnd_Alias SOK_SITEA   = /usr/sbin/crm_attribute -n hana_<sid>_site_srHook_<siteA> -
v SOK   -t crm_config -s SAPHanaSR
Cmnd_Alias SFAIL_SITEA = /usr/sbin/crm_attribute -n hana_<sid>_site_srHook_<siteA> -
v SFAIL -t crm_config -s SAPHanaSR
Cmnd_Alias SOK_SITEB   = /usr/sbin/crm_attribute -n hana_<sid>_site_srHook_<siteB> -
v SOK   -t crm_config -s SAPHanaSR
```

```
Cmnd_Alias SFAIL_SITEB = /usr/sbin/crm_attribute -n hana_<sid>_site_srHook_<siteB> -
v SFAIL -t crm_config -s SAPHanaSR
<sidadm> ALL=(ALL) NOPASSWD: SOK_SITEA, SFAIL_SITEA, SOK_SITEB, SFAIL_SITEB
```

9 Configuring the Cluster



This chapter describes the configuration of the cluster software SUSE Linux Enterprise High Availability Extension, which is part of SUSE Linux Enterprise Server for SAP Applications, and SAP HANA Database Integration.

ACTIONS

1. Basic Cluster Configuration
2. Configure Cluster Properties and Resources

9.1 Installation

AWS "SLES for SAP" AMIs already have all High Availability Extension packages installed.

It is recommended to update all packages to make sure that the latest revision of the cluster packages and AWS agents are installed.

EXAMPLE 30: UPDATING SUSE LINUX ENTERPRISE SERVER WITH ALL LATEST PATCHES

```
suse01:~> zypper update
```

9.2 Configuring the Basic Cluster

The first step is to set up the basic cluster framework.

9.2.1 Configuring Corosync

By default, the cluster service (pacemaker) is disabled and not set to start during boot. Thus at this point the cluster should not be running. However, if you previously configured pacemaker and it is running, proceed with a "stop" by using the following command:

EXAMPLE 31: STOPPING THE CLUSTER

```
suse01:~ # systemctl stop pacemaker
```

The cluster service (pacemaker) status can be checked with:

EXAMPLE 32: CHECKING CLUSTER STATUS

```
suse01:~ # systemctl status pacemaker
```

9.2.2 Creating Keys

On Node 1, generate a corosync secret key used to encrypt all cluster communication:

EXAMPLE 33: GENERATING COROSYNC SECURITY KEYS

```
suse01:~# corosync-keygen
```

A new key file will be created on */etc/corosync/authkey*, and this file needs to be copied to the same location on Node 2. After generating and transferring the key file to the second node, verify that permissions and ownerships on both nodes are the same:

EXAMPLE 34: CHECKING PERMISSIONS AND OWNERSHIP FOR COROSYNC KEY FILE

```
suse01:~ # ls -l /etc/corosync/authkey
-r-------- 1 root root 128 Oct 23 10:51 /etc/corosync/authkey
```

9.2.3 Creating the Corosync Configuration File

The corosync configuration will leverage both IP addresses associated to each cluster node. The two IP configurations will use the second IP if the primary IP addresses for the two node cluster are no longer able to communicate with each other.

All cluster nodes are required to have a local configuration file *"/etc/corosync/corosync.conf"* where 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.

AWS requires a specific corosync configuration, which can be structured as the example below.

Note

When using the following configuration as an example for the file */etc/corosync/corosync.conf*, replace the IP addresses from the file below.

EXAMPLE 35: SAMPLE COROSYC.CONF FILE

```
# Read the corosync.conf.5 manual page
totem {
   version: 2
   rrp_mode: passive
   token: 30000
   consensus: 36000
   token_retransmits_before_loss_const: 6
   secauth: on
   crypto_hash: sha1
   crypto_cipher: aes256
   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-a
     # redundant ring
     ring1_addr: ip-node-1-b
     nodeid: 1
   }
   node {
```

```
      ring0_addr: ip-node-2-a
      # redundant ring
      ring1_addr: ip-node-2-b
      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-a*, *ip-node-1-b*, *ip-node-2-a*, *ip-node-2-b* and *ip-local-node* from the above sample file.

- **ip-local-node**: Use the IP address of the node where the file is being configured. This IP will be different between cluster nodes.
- **ip-node-1-a**: Primary IP address of cluster node node-1
- **ip-node-1-b**: Secondary IP address of cluster node node-1
- **ip-node-2-a**: Primary IP address of cluster node node-2
- **ip-node-2-b**: Secondary IP address of cluster node node-2

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.

Note

Remember to change the password of the user hacluster.

9.2.4 Checking the Cluster for the First Time

Now it is time to check and start the cluster for the first time on both nodes.

EXAMPLE 36: STARTING THE CLUSTER ON BOTH CLUSTER NODES

```
suse01:~ # systemctl status pacemaker
suse02:~ # systemctl status pacemaker
```

```
suse01:~ # systemctl start pacemaker
suse02:~ # systemctl start pacemaker
```

Check the cluster status with `crm_mon`. We use the option `-r` to also see resources which may be configured but stopped. But at this stage `crm_mon` is expected to display no services.

EXAMPLE 37: CHECKING CLUSTER STATUS USING CRM_MON

```
# crm_mon -r
```

The command will show the "empty" cluster and will print something like the computer output shown below. The most interesting information for now is that there are two nodes in the status "online", and the message "partition with quorum".

EXAMPLE 38: CLUSTER STATUS AFTER FIRST START

```
Stack: corosync
Current DC: prihana (version 1.1.19+20181105.ccd6b5b10-3.19.1-1.1.19+20181105.ccd6b5b10)
 - partition with quorum
Last updated: Mon Sep 28 18:36:16 2020
Last change: Mon Sep 28 18:36:09 2020 by root via crm_attribute on suse01

2 nodes configured

2 nodes configured
0 resources configured

Online: [ suse01 suse02 ]

No resources
```

Corosync's redundant ring configuration can be checked with the following command:

EXAMPLE 39: COROSYNC REDUNDANT RING STATUS

```
corosync-cfgtool -s
```

This will display a result like the following one for a cluster node with redundant corosync rings and IP addresses 172.16.100.179 and 172.16.100.138:

```
Printing ring status.
Local node ID 1
RING ID 0
 id = 172.16.100.179
 status = ring 0 active with no faults
RING ID 1
 id = 172.16.100.138
 status = ring 1 active with no faults
```

Note

It is not recommended to automatically rejoin a node to a cluster after a system crash with a reboot. A full inspection and a root cause analysis of the crash is highly recommended before rejoining the cluster.

9.3 Configuring Cluster Properties and Resources

This section describes how to configure constraints, resources, bootstrap and STONITH using the `crm configure` shell command as described in section *Configuring and Managing Cluster Resources (Command Line)* of the SUSE Linux Enterprise High Availability Extension documentation.

Use the command `crm` to add the objects to CRM. Copy the following examples to a local file, edit the file and then load the configuration to the CIB:

```
suse01:~ # vi crm-fileXX
suse01:~ # crm configure load update crm-fileXX
```

9.3.1 Cluster Bootstrap and More

The first example defines the cluster bootstrap options, the resource and operation defaults.

```
suse01:~ # vi crm-bs.txt
# enter the following to the file crm-bs.txt
property $id="cib-bootstrap-options" \
   stonith-enabled="true" \
   stonith-action="off" \
   stonith-timeout="600s"
rsc_defaults $id="rsc-options" \
   resource-stickiness="1000" \
   migration-threshold="5000"
op_defaults $id="op-options" \
   timeout="600"
```

Note

In some older SUSE versions, the parameter *stonith-action* may require a change to `stonith-action="poweroff"`.

The setting *off* forces the EC2 STONITH agent to shut down the EC2 instance in case of fencing operation. This is desirable to avoid split brain scenarios on the AWS platform.

Now, add the configuration to the cluster:

```
suse01:~ # crm configure load update crm-bs.txt
```

9.3.2 STONITH Device

The next configuration part defines an AWS EC2 STONITH resource.

```
suse01::~ # vi aws-stonith.txt
# enter the following to the file aws-stonith.txt
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 \
    meta target-role=Started \
    params tag=pacemaker profile=cluster pcmk_delay_max=15
```

The *"tag=pacemaker"* entry needs to match the tag chosen for the EC2 instances. The value for this tag contains the host name returned by the `uname -n` command. The name of the profile ("cluster" in this example) needs to match the previously configured profile in the AWS CLI.

Name this file for example *aws-stonith.txt* and add it to the configuration. The following command needs to be issued as *root* user:

```
suse01:~ # crm configure load update aws-stonith.txt
```

A working STONITH method is mandatory to run a supported SUSE cluster on AWS.

Note

Make sure to execute the STONITH tests as outlined in section *Troubleshooting* of this document to verify STONITH on both nodes.

Note

The STONITH agent currently requires the pacemaker tag on the instance to be in lowercase format. i.e a hostname of 'NODE1', should have the pacemaker tag on the instance set to 'node1'. This will be updated in the future.

9.3.3 Configuring the Overlay IP address

This step requires the Overlay IP address and the resource IDs of the AWS VPC Route Table(s). Create a file with the following content:

```
suse01:~ # vi aws-move-ip.txt
# enter the following to the file aws-move-ip.txt
primitive res_AWS_IP ocf:suse:aws-vpc-move-ip \
   params ip=overlay-ip-address 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
```

Replace the following individual parameters with the appropriate values:

- *overlay-ip-address* : the Overlay IP address used
- *rtb-table* : The AWS VPC Route Table(s) resource ids - if using more than one VPC Route Table use comma (,) as a separator (see below).
- *interface* : The Linux' network interface identificator
- *profile* : The name of the profile (cluster in this example) needs to match the previously configured profile in the AWS CLI.

Load this file into the cluster configuration by issuing the following command as *superuser*:

```
suse01:~ # crm configure load update aws-move-ip.txt
```

Optionally, it is possible to specify multiple routing tables in the primitive configuration separated by a comma (,), as shown in the following example:

```
suse01:~ # vi aws-move-ip.txt
# enter the following to the file aws-move-ip.txt
primitive res_AWS_IP ocf:suse:aws-vpc-move-ip \
   params ip=overlay-ip-address routing_table=rtb-table-1,rtb-table-2,rtb-table-N
 interface=eth0 profile=cluster \
   op start interval=0 timeout=180 \
   op stop interval=0 timeout=180 \
   op monitor interval=60 timeout=60
```

Note

Make sure to execute the IP tests as outlined in section *Troubleshooting* of this document to verify them on both nodes. Checking the configuration for potential problems at current point in time will increase the chances to launch the cluster successfully.

9.3.4 SAPHanaTopology

Next, define the group of resources needed, before the HANA instances can be started. Prepare the changes in a text file, for example *crm-saphanatop.txt*, and load it with the command:

```
crm configure load update crm-saphanatop.txt
```

```
# vi crm-saphanatop.txt
# enter the following to crm-saphanatop.txt
primitive rsc_SAPHanaTopology_HA1_HDB10 ocf:suse:SAPHanaTopology \
        op monitor interval="10" timeout="600" \
        op start interval="0" timeout="600" \
        op stop interval="0" timeout="300" \
        params SID="HA1" InstanceNumber="10"
clone cln_SAPHanaTopology_HA1_HDB10 rsc_SAPHanaTopology_HA1_HDB10 \
        meta clone-node-max="1" interleave="true"
```

Additional information about all parameters can be found with the command:

```
man ocf_suse_SAPHanaTopology
```

Again, add the configuration to the cluster.

```
suse01:~ # crm configure load update crm-saphanatop.txt
```

The most important parameters here are SID and InstanceNumber, which are quite self-explaining in the SAP context. Beside these parameters, the timeout values or the operations (start, monitor, stop) are typical tuneables.

9.3.5 SAPHana

Next, define the group of resources needed, before the HANA instances can be started. Edit the changes in a text file, for example *crm-saphana.txt*, and load it with the command:

```
crm configure load update crm-saphana.txt
```

TABLE 2: TYPICAL RESOURCE AGENT PARAMETER SETTINGS FOR DIFFERENT SCENARIOS

Parameter	Performance Optimized	Cost Optimized	Multi-Tier
PREFER_SITE_TAKEOVER	true	false	false / true
AUTOMATED_REGISTER	false / true	false / true	false
DUPLICATE_PRIMARY_TIMEOUT	7200	7200	7200

TABLE 3: DESCRIPTION OF IMPORTANT RESOURCE AGENT PARAMETERS

Parameter	Description
PREFER_SITE_TAKEOVER	Defines whether RA should prefer to takeover to the secondary instance instead of restarting the failed primary locally.
AUTOMATED_REGISTER	Defines whether a former primary should be automatically registered to be secondary of the new primary. With this parameter you can adapt the level of system replication automation. If set to `false`, the former primary must be manually registered. The cluster will not start this SAP HANA RDBMS until it is registered to avoid double primary up situations.
DUPLICATE_PRIMARY_TIMEOUT	Time difference needed between two primary time stamps if a dual-primary situation occurs. If the time difference is less than the time gap, than the cluster hold one or both instances in a "WAITING" status. This is to give an administrator the chance to react on a fail-over. If the complete node of the former primary crashed, the former primary will be registered after the time difference is passed. If "only" the SAP HANA RDBMS has crashed, then the former primary will be registered immediately. After this registration to the new primary all data will be overwritten by the system replication.

Additional information about all parameters can be found with the command:

`man ocf_suse_SAPHana`

```
# vi crm-saphana.txt
# enter the following to crm-saphana.txt
primitive rsc_SAPHana_HA1_HDB10 ocf:suse:SAPHana \
        op start interval="0" timeout="3600" \
        op stop interval="0" timeout="3600" \
        op promote interval="0" timeout="3600" \
        op monitor interval="60" role="Master" timeout="700" \
```

```
        op monitor interval="61" role="Slave" timeout="700" \
        params SID="HA1" InstanceNumber="10" PREFER_SITE_TAKEOVER="true" \
        DUPLICATE_PRIMARY_TIMEOUT="7200" AUTOMATED_REGISTER="false"
ms msl_SAPHana_HA1_HDB10 rsc_SAPHana_HA1_HDB10 \
        meta clone-max="2" clone-node-max="1" interleave="true"
```

Add the configuration to the cluster.

```
suse01:~ # crm configure load update crm-saphana.txt
```

The most important parameters here are again SID and InstanceNumber. Beside these parameters, the timeout values for the operations (start, promote, monitors, stop) are typical tuneables.

9.3.6 Constraints

Two constraints are organizing the correct placement of the virtual IP address for the client database access and the start order between the two resource agents SAPHana and SAPHanaTopology.

The AWS IP agent needs to operate on the same node as the SAP HANA Master database. A constraint forces it to be on the same node.

```
# vi crm-cs.txt
# enter the following to crm-cs.txt

colocation col_saphana_ip_HA1_HDB10 2000: res_AWS_IP:Started \
    msl_SAPHana_HA1_HDB10:Master
order ord_SAPHana_HA1_HDB10 Optional: cln_SAPHanaTopology_HA1_HDB10 \
    msl_SAPHana_HA1_HDB10
```

Add this file to the configuration. The following command needs to be issued as *superuser*. It uses the file name *crm-cs.txt*:

```
suse01:~ # crm configure load update crm-cs.txt
```

9.3.7 Active/Active Read-Enabled Scenario

This step is optional. If you have an active/active SAP HANA system replication with a read-enabled secondary, it is possible to integrate the needed second Overlay IP address into the cluster. This is done by adding a second Overlay IP address resource and a location constraint binding the address to the secondary site.

```
# vi crm-re.txt
```

```
# enter the following to crm-re.txt
primitive res_AWS_IP_readenabled ocf:suse:aws-vpc-move-ip \
   params ip=readenabled-overlay-ip-address 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
colocation col_saphana_ip_HA1_HDB10_readenabled 2000: \
    res_AWS_IP_readenabled:Started msl_SAPHana_HA1_HDB10:Slave
```

9.3.8 Cluster Status After Configuration

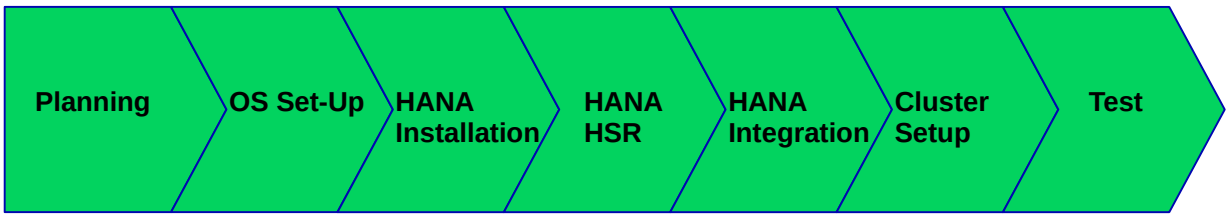
Now that the cluster has been configured, the basic it should have two online nodes, and six resources. If you configured a second Overlay IP for the read enabled replica, then the cluster will display seven resources.

The cluster status can be checked with `crm status` command:

```
suse01:~ # crm status
Stack: corosync
Current DC: prihana (version 1.1.19+20181105.ccd6b5b10-3.19.1-1.1.19+20181105.ccd6b5b10)
 - partition with quorum
Last updated: Tue Sep 29 16:15:51 2020
Last change: Tue Sep 29 16:15:05 2020 by root via crm_attribute on prihana

2 nodes configured
6 resources configured

Online: [ suse01 suse02 ]

Full list of resources:

 res_AWS_STONITH (stonith:external/ec2): Started suse01
 res_AWS_IP (ocf::suse:aws-vpc-move-ip): Started suse01
 Clone Set: cln_SAPHanaTopology_HDB_HDB00 [rsc_SAPHanaTopology_HDB_HDB00]
     Started: [ suse01 suse02 ]
 Master/Slave Set: msl_SAPHana_HDB_HDB00 [rsc_SAPHana_HDB_HDB00]
     Masters: [ suse01 ]
     Slaves: [ suse02 ]
```

The above example shows that the Overlay IP resource (res_AWS_IP) is "Started" on node suse01, along with SAPHanaTopology resource (cln_SAPHanaTopology_HA1_HDB10) running on both cluster nodes, and Master/Slave SAPHana (msl_SAPHana_HA1_HDB10), which in the above example is Master (Primary) on node suse01, and Secondary on node suse02.

10 Testing the Cluster



The lists of tests will be enhanced with future updates of this document.

As with any cluster testing is crucial. Make sure that all test cases derived from customer expectations are implemented and fully passed. Otherwise the project is likely to fail in production.

The test prerequisite, if not described differently, is always that both nodes are booted, normal members of the cluster, and the HANA RDBMS is running. The system replication is in sync (SOK).

10.1 Test Cases for Semi Automation

In the following test descriptions we assume the following values:

`PREFER_SITE_TAKEOVER="true"` and `AUTOMATED_REGISTER="false"`

Note

The following tests are designed to run in a sequence. They depend on the exit state of the proceeding tests.

10.1.1 Test: Stop Primary Database on Availability Zone A (Node 1)

EXAMPLE 40: TEST STOP_PRIMARY_SITE_A

COMPONENT:

- Primary Database

DESCRIPTION:

- The primary HANA database is stopped during normal cluster operation.

TEST PROCEDURE:

1. Stop the primary HANA database gracefully as *<sid>adm*.

```
suse01# HDB stop
```

RECOVERY PROCEDURE:

1. Manually register the old primary (on node 1) with the new primary after takeover (on node 2) as *<sid>adm*.

```
suse01# hdbnsutil -sr_register --remoteHost=suse02 --remoteInstance=10 \
          --replicationMode=sync --operationMode=logreplay \
          --name=WDF
```

2. Restart the HANA database (now secondary) on node 1 as *root*.

```
suse01# crm resource refresh rsc_SAPHana_HA1_HDB10 suse01
```

EXPECTED:

1. The cluster detects the stopped primary HANA database (on node 1) and marks the resource failed.
2. The cluster promotes the secondary HANA database (on node 2) to take over as primary.
3. The cluster migrates the IP address to the new primary (on node 2).
4. After some time the cluster shows the sync_state of the stopped primary (on node 1) as SFAIL.
5. Because of AUTOMATED_REGISTER = "false" the cluster does not restart the failed HANA database or register it against the new primary.
6. After the manual register and resource refresh, the system replication pair is marked as in sync (SOK).
7. The cluster "failed actions" are cleaned up after following the recovery procedure.

10.1.2 Test: Stop Primary Database on Availability Zone B (Node 2)

EXAMPLE 41: TEST STOP_PRIMARY_DB_SITE_B

Component:

Primary Database

Description:

The primary HANA database is stopped during normal cluster operation.

TEST PROCEDURE:

1. Stop the database gracefully as *<sid>adm*.

```
suse02# HDB stop
```

RECOVERY PROCEDURE:

1. Manually register the old primary (on node 2) with the new primary after takeover (on node 1) as *<sid>adm*.

```
suse02# hdbnsutil -sr_register --remoteHost=suse01 --remoteInstance=10 \
               --replicationMode=sync --operationMode=logreplay \
               --name=ROT
```

2. Restart the HANA database (now secondary) on node 1 as *root*.

```
suse02# crm resource refresh rsc_SAPHana_HA1_HDB10 suse02
```

EXPECTED:

1. The cluster detects the stopped primary HANA database (on node 2) and marks the resource failed.
2. The cluster promotes the secondary HANA database (on node 1) to take over as primary.
3. The cluster migrates the IP address to the new primary (on node 1).
4. After some time, the cluster shows the sync_state of the stopped primary (on node 2) as SFAIL.
5. Because of AUTOMATED_REGISTER = "false" the cluster does not restart the failed HANA database or register it against the new primary.
6. After the manual register and resource refresh, the system replication pair is marked as in sync (SOK).
7. The cluster "failed actions" are cleaned up after following the recovery procedure.

10.1.3 Test: Crash Primary Database on Availability Zone A (Node 1)

EXAMPLE 42: TEST CRASH_PRIMARY_DB_SITE_A

Component:

Primary Database

Description:

Simulate a complete breakdown of the primary database system.

TEST PROCEDURE:

1. Kill the primary database system using signals as *<sid>adm*.

```
suse01# HDB kill-9
```

RECOVERY PROCEDURE:

1. Manually register the old primary (on node 1) with the new primary after takeover (on node 2) as *<sid>adm*.

```
suse01# hdbnsutil -sr_register --remoteHost=suse02 --remoteInstance=10 \
               --replicationMode=sync  --operationMode=logreplay \
               --name=WDF
```

2. Restart the HANA database (now secondary) on node 1 as *root*.

```
suse01# crm resource refresh rsc_SAPHana_HA1_HDB10 suse01
```

EXPECTED:

1. The cluster detects the stopped primary HANA database (on node 1) and marks the resource failed.
2. The cluster promotes the secondary HANA database (on node 2) to take over as primary.
3. The cluster migrates the IP address to the new primary (on node 2).
4. After some time, the cluster shows the sync_state of the stopped primary (on node 1) as SFAIL.
5. Because of AUTOMATED_REGISTER="false" the cluster does not restart the failed HANA database or register it against the new primary.

6. After the manual register and resource refresh, the system replication pair is marked as in sync (SOK).
7. The cluster "failed actions" are cleaned up after following the recovery procedure.

10.1.4 Test: Crash Primary Database on Availability Zone B (Node 2)

EXAMPLE 43: TEST CRASH_PRIMARY_DB_SITE_B

Component:

Primary Database

Description:

Simulate a complete breakdown of the primary database system.

TEST PROCEDURE:

1. Kill the primary database system using signals as *<sid>adm*.

```
suse02# HDB kill-9
```

RECOVERY PROCEDURE:

1. Manually register the old primary (on node 2) with the new primary after takeover (on node 1) as *<sid>adm*.

```
suse02# hdbnsutil -sr_register --remoteHost=suse01 --remoteInstance=10 \
               --replicationMode=sync  --operationMode=logreplay \
               --name=ROT
```

2. Restart the HANA database (now secondary) on node 1 as *root*.

```
suse02# crm resource refresh rsc_SAPHana_HA1_HDB10 suse02
```

EXPECTED:

1. The cluster detects the stopped primary HANA database (on node 2) and marks the resource failed.
2. The cluster promotes the secondary HANA database (on node 1) to take over as primary.

3. The cluster migrates the IP address to the new primary (on node 1).
4. After some time, the cluster shows the sync_state of the stopped primary (on node 2) as SFAIL.
5. Because of AUTOMATED_REGISTER = "false" the cluster does not restart the failed HANA database or register it against the new primary.
6. After the manual register and resource refresh, the system replication pair is marked as in sync (SOK).
7. The cluster "failed actions" are cleaned up after following the recovery procedure.

10.1.5 Test: Crash Primary Node on Availability Zone A (Node 1)

EXAMPLE 44: TEST CRASH_PRIMARY_NODE_SITE_A

Component:

Cluster node of primary site

Description:

Simulate a crash of the primary site node running the primary HANA database.

TEST PROCEDURE:

1. Crash the primary node by sending a 'fast-reboot' system request.

```
suse01# echo 'b' > /proc/sysrq-trigger
```

RECOVERY PROCEDURE:

1. AWS infrastructure has stopped the fenced instance. Restart it with AWS console or AWS CLI tools. Execute the following command after the instance has booted.
2. Start the cluster framework.

```
suse01# systemctl start pacemaker
```

3. Manually register the old primary (on node 1) with the new primary after takeover (on node 2) as *<sid>adm*.

```
suse01# hdbnsutil -sr_register --remoteHost=suse02 --remoteInstance=10 \
```

```
                --replicationMode=sync  --operationMode=logreplay \
                --name=WDF
```

4. Restart the HANA database (now secondary) on node 1 as *root*.

```
suse01# crm resource refresh rsc_SAPHana_HA1_HDB10 suse01
```

EXPECTED:

1. The cluster detects the failed node (node 1) and declares it UNCLEAN and sets the secondary node (node 2) to status "partition with quorum".
2. The cluster fences the failed node (node 1).
3. The cluster declares the failed node (node 1) OFFLINE.
4. The cluster promotes the secondary HANA database (on node 2) to take over as primary.
5. The cluster migrates the IP address to the new primary (on node 2).
6. After some time, the cluster shows the sync_state of the stopped primary (on node 2) as SFAIL.
7. Because of AUTOMATED_REGISTER = "false" the cluster does not restart the failed HANA database or register it against the new primary.
8. After the manual register and resource refresh, the system replication pair is marked as in sync (SOK).
9. The cluster "failed actions" are cleaned up after following the recovery procedure.

10.1.6 Test: Crash Primary Node on Availability Zone B (Node 2)

EXAMPLE 45: TEST CRASH_PRIMARY_NODE_SITE_B

Component:

Cluster node of secondary site

Description:

Simulate a crash of the secondary site node running the primary HANA database.

TEST PROCEDURE:

1. Crash the secondary node by sending a 'fast-reboot' system request.

```
suse02# echo 'b' > /proc/sysrq-trigger
```

RECOVERY PROCEDURE:

1. AWS infrastructure has stopped the fenced instance. Restart it with AWS console or AWS CLI tools. Execute the following command after the instance has booted.
2. Start the cluster Framework

```
suse02# systemctl start pacemaker
```

3. Manually register the old primary (on node 2) with the new primary after takeover (on node 1) as *<sid>adm*.

```
suse02# hdbnsutil -sr_register --remoteHost=suse01 --remoteInstance=10 \
               --replicationMode=sync  --operationMode=logreplay \
               --name=ROT
```

4. Restart the HANA database (now secondary) on node 2 as *root*.

```
suse02# crm resource refresh rsc_SAPHana_HA1_HDB10 suse02
```

EXPECTED:

1. The cluster detects the failed secondary node (node 2) and declares it UNCLEAN and sets the primary node (node 1) to status "partition with quorum".
2. The cluster fences the failed secondary node (node 2).
3. The cluster declares the failed secondary node (node 2) OFFLINE.
4. The cluster promotes the secondary HANA database (on node 1) to take over as primary.
5. The cluster migrates the IP address to the new primary (on node 1).
6. After some time, the cluster shows the sync_state of the stopped secondary (on node 2) as SFAIL.
7. Because of AUTOMATED_REGISTER = "false" the cluster does not restart the failed HANA database or register it against the new primary.

8. After the manual register and resource refresh, the system replication pair is marked as in sync (SOK).
9. The cluster "failed actions" are cleaned up after following the recovery procedure.

10.1.7 Test: Stop Secondary Database on Availability Zone B (Node 2)

EXAMPLE 46: TEST STOP_SECONDARY_DB_SITE_B

Component:

Secondary HANA database

Description:

The secondary HANA database is stopped during normal cluster operation.

TEST PROCEDURE:

1. Stop the secondary HANA database gracefully as *<sid>adm*.

```
suse02# HDB stop
```

RECOVERY PROCEDURE:

1. Refresh the failed resource status of the secondary HANA database (on node 2) as *root*.

```
suse02# crm resource refresh rsc_SAPHana_HA1_HDB10 suse02
```

EXPECTED:

1. The cluster detects the stopped secondary database (on node 2) and marks the resource failed.
2. The cluster detects the broken system replication and marks it as failed (SFAIL).
3. The cluster restarts the secondary HANA database on the same node (node 2).
4. The cluster detects that the system replication is in sync again and marks it as ok (SOK).
5. The cluster "failed actions" are cleaned up after following the recovery procedure.

10.1.8 Test: Crash Secondary Database on Availability Zone B (Node 2)

EXAMPLE 47: TEST CRASH_SECONDARY_DB_SITE_B

Component:

Secondary HANA database

Description:

Simulate a complete breakdown of the secondary database system.

TEST PROCEDURE:

1. Kill the secondary database system using signals as *<sid>adm*.

```
suse02# HDB kill-9
```

RECOVERY PROCEDURE:

1. Clean up the failed resource status of the secondary HANA database (on node 2) as *root*.

```
suse02# crm resource refresh rsc_SAPHana_HA1_HDB10 suse02
```

EXPECTED:

1. The cluster detects the stopped secondary database (on node 2) and marks the resource failed.
2. The cluster detects the broken system replication and marks it as failed (SFAIL).
3. The cluster restarts the secondary HANA database on the same node (node 2).
4. The cluster detects that the system replication is in sync again and marks it as ok (SOK).
5. The cluster "failed actions" are cleaned up after following the recovery procedure.

10.1.9 Test: Crash Secondary Node on Availability Zone B (Node2)

EXAMPLE 48: TEST CRASH_SECONDARY_NODE_SITE_B

Component:

Cluster node of secondary site

Description:

Simulate a crash of the secondary site node running the secondary HANA database.

TEST PROCEDURE:

1. Crash the secondary node by sending a 'fast-reboot' system request.

```
suse02# echo 'b' > /proc/sysrq-trigger
```

RECOVERY PROCEDURE:

1. AWS infrastructure has stopped the fenced instance. Restart it with AWS console or AWS CLI tools. Execute the following command after the instance has booted.
2. Start the cluster framework.

```
suse02# systemctl start pacemaker
```

EXPECTED:

1. The cluster detects the failed secondary node (node 2) and declares it UNCLEAN and sets the primary node (node 1) to status "partition with quorum".
2. The cluster fences the failed secondary node (node 2).
3. The cluster declares the failed secondary node (node 2) OFFLINE.
4. After some time, the cluster shows the sync_state of the stopped secondary (on node 2) as SFAIL.
5. When the fenced node (node 2) rejoins the cluster, the former secondary HANA database is started automatically.
6. The cluster detects that the system replication is in sync again and marks it as ok (SOK).

10.2 Test: Failure of Replication LAN

Component: Replication LAN

Description: This test is not applicable to AWS. There is no separate replication LAN.

10.3 Test Cases for Full Automation

In the following test descriptions we assume `PREFER_SITE_TAKEOVER="true"' and `AUTOMATED_REGISTER="true"`.

Note

The following tests are designed to run in a sequence. They depend on the exit state of the proceeding tests.

10.3.1 Test: Stop Primary Database on Site A

EXAMPLE 49: TEST STOP_PRIMARY_DB_SITE_A

COMPONENT:

- Primary Database

DESCRIPTION:

- The primary HANA database is stopped during normal cluster operation.

TEST PROCEDURE:

- Stop the primary HANA database gracefully as *<sid>adm*.

```
suse01# HDB stop
```

RECOVERY PROCEDURE:

1. Not needed, everything is automated
2. Refresh the cluster resources on node 1 as *root*.

```
suse01# crm resource refresh rsc_SAPHana_HA1_HDB10 suse01
```

EXPECTED:

1. The cluster detects the stopped primary HANA database (on node 1) and marks the resource failed.
2. The cluster promotes the secondary HANA database (on node 2) to take over as primary.
3. The cluster migrates the IP address to the new primary (on node 2).
4. After some time, the cluster shows the sync_state of the stopped primary (on node 1) as SFAIL.
5. Because of AUTOMATED_REGISTER = "true" the cluster does restart the failed HANA database and register it against the new primary.

6. After the automated register and resource refresh, the system replication pair is marked as in sync (SOK).
7. The cluster "failed actions" are cleaned up after following the recovery procedure.

10.3.2 Test: Crash the Primary Node on Site B (Node 2)

EXAMPLE 50: TEST CRASH_PRIMARY_NODE_SITE_B

COMPONENT:

- Cluster node of site B

DESCRIPTION:

- Simulate a crash of the site B node running the primary HANA database.

TEST PROCEDURE:

- Crash the secondary node by sending a 'fast-reboot' system request.

```
suse02# echo 'b' > /proc/sysrq-trigger
```

RECOVERY PROCEDURE:

- Start the cluster framework.

```
suse02# systemctl start pacemaker
```

- Refresh the cluster resources on node 2 as *root*.

```
suse02# crm resource refresh rsc_SAPHana_HA1_HDB10 suse02
```

EXPECTED:

1. The cluster detects the failed primary node (node 2) and declares it UNCLEAN and sets the primary node (node 2) to status "partition with quorum".
2. The cluster fences the failed primary node (node 2).
3. The cluster declares the failed primary node (node 2) OFFLINE.
4. The cluster promotes the secondary HANA database (on node 1) to take over as primary.
5. The cluster migrates the IP address to the new primary (on node 1).

6. After some time, the cluster shows the sync_state of the stopped secondary (on node 2) as SFAIL.
7. When the fenced node (node 2) rejoins the cluster, the former primary becomes a secondary.
8. Because of AUTOMATED_REGISTER = "true" the cluster does restart the failed HANA database and register it against the new primary.
9. The cluster detects that the system replication is in sync again and marks it as ok (SOK).

11 Administration

11.1 Dos and Don’ts

In your project, you should:

- define STONITH before adding other resources to the cluster.
- do intensive testing.
- tune the timeouts of operations of SAPHana and SAPHanaTopology.
- start with the values PREFER_SITE_TAKEOVER = ”true”, AUTOMATED_REGISTER = ”false” and DUPLICATE_PRIMARY_TIMEOUT = ”7200”.

In your project, avoid:

- rapidly changing/changing back cluster configuration, such as setting nodes to standby and online again or stopping/starting the master/slave resource.
- creating a cluster without proper time synchronization or unstable name resolutions for hosts, users and groups.
- adding location rules for the clone, master/slave or IP resource. Only location rules mentioned in this setup guide are allowed.
- "migrating" or "moving" resources in crm-shell, HAWK or other tools because this would add client-prefer location rules. Thus, these activities are completely forbidden.

11.2 Monitoring and Tools

You can use the High Availability Web Console (HAWK), SAP HANA Studio and different command line tools for cluster status requests.

11.2.1 HAWK – Cluster Status and More

You can use an Internet browser to check the cluster status.

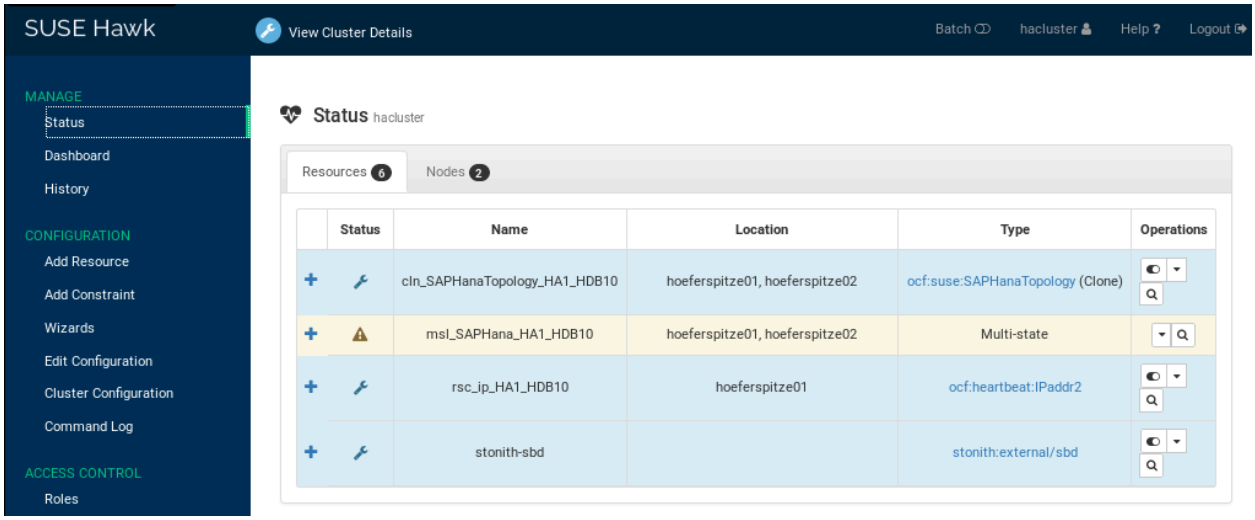


FIGURE 10: CLUSTER STATUS IN HAWK

If you set up the cluster using ha-cluster-init and you have installed all packages as described above, your system will provide a very useful Web interface. You can use this graphical Web interface to get an overview of the complete cluster status, perform administrative tasks or configure resources and cluster bootstrap parameters. Read the product manuals for a complete documentation of this powerful user interface.

11.2.2 SAP HANA Studio

Database-specific administration and checks can be done with SAP HANA studio.

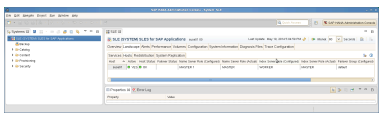


FIGURE 11: SAP HANA STUDIO – LANDSCAPE

11.2.3 Cluster Command Line Tools

A simple overview can be obtained by calling `crm_mon`. Using the option `-r` shows also stopped but already configured resources. Option `-1` tells `crm_mon` to output the status once instead of periodically.

```
Stack: corosync
Current DC: suse01 (version 1.1.19+20181105.ccd6b5b10-3.19.1-1.1.19+20181105.ccd6b5b10) -
 partition with quorum
Last updated: Mon Sep 28 18:36:16 2020
Last change: Mon Sep 28 18:36:09 2020 by root via crm_attribute on prihana

2 nodes configured
6 resources configured

Online: [ suse01 suse02 ]

Full list of resources:

 res_AWS_STONITH (stonith:external/ec2): Started suse01
 res_AWS_IP (ocf::suse:aws-vpc-move-ip): Started suse01
 Clone Set: cln_SAPHanaTopology_HDB_HDB00 [rsc_SAPHanaTopology_HDB_HDB00]
     Started: [ suse01 suse02 ]
 Master/Slave Set: msl_SAPHana_HDB_HDB00 [rsc_SAPHana_HDB_HDB00]
     Masters: [ suse01 ]
     Slaves: [ suse02 ]
```

See the manual page crm_mon(8) for details.

11.2.4 SAPHanaSR Command Line Tools

To show some SAPHana or SAPHanaTopology resource agent internal values, you can call the program `SAPHanaSR-showAttr`. The internal values, the storage location and their parameter names may change in the next versions of this document. The command `SAPHanaSR-showAttr` will always fetch the values from the correct storage location.

Do not use cluster commands like `crm_attribute` to fetch the values directly from the cluster. If you use such commands, your methods will be broken when you need to move an attribute to a different storage place or even out of the cluster. At first, `SAPHanaSR-showAttr` is a test program only and should not be used for automated system monitoring.

```
suse01:~ # SAPHanaSR-showAttr
Host \ Attr clone_state remoteHost roles       ... site    srmode sync_state ...
--------------------------------------------------------------------------------
```

```
 suse01      PROMOTED    suse02     4:P:master1:... WDF      sync  PRIM       ...
 suse02      DEMOTED     suse01     4:S:master1:... ROT      sync  SOK        ...
```

`SAPHanaSR-showAttr` also supports other output formats such as **script**. The script format is intended to allow running filters. The SAPHanaSR package beginning with version 0.153 also provides a filter engine `SAPHanaSR-filter`. In combination of `SAPHanaSR-showAttr` with output format script and `SAPHanaSR-filter` you can define effective queries:

```
suse01:~ # SAPHanaSR-showAttr --format=script | \
   SAPHanaSR-filter --search='remote'
Mon Nov 11 20:55:45 2019; Hosts/suse01/remoteHost=suse02
Mon Nov 11 20:55:45 2019; Hosts/suse02/remoteHost=suse01
```

`SAPHanaSR-replay-archive` can help to analyze the SAPHanaSR attribute values from `hb_report` (`crm_report`) archives. This allows post mortem analyses.

In our example, the administrator killed the primary SAP HANA instance using the command `HDB kill-9`. This happened around 9:10 pm.

```
suse01:~ # hb_report -f 19:00
INFO: suse01# The report is saved in ./hb_report-1-11-11-2019.tar.bz2
INFO: suse01# Report timespan: 11/11/19 19:00:00 - 11/11/19 21:05:33
INFO: suse01# Thank you for taking time to create this report.
suse01:~ # SAPHanaSR-replay-archive --format=script \
    ./hb_report-1-11-11-2019.tar.bz2 | \
    SAPHanaSR-filter --search='roles' --filterDouble
Mon Nov 11 20:38:01 2019; Hosts/suse01/roles=4:P:master1:master:worker:master
Mon Nov 11 20:38:01 2019; Hosts/suse02/roles=4:S:master1:master:worker:master
Mon Nov 11 21:11:37 2019; Hosts/suse01/roles=1:P:master1::worker:
Mon Nov 11 21:12:43 2019; Hosts/suse02/roles=4:P:master1:master:worker:master
```

In the above example the attributes indicate that at the beginning suse01 was running primary (4:P) and suse02 was running secondary (4:S).

At 21:11 (CET) suddenly the primary on suse01 died - it was falling down to 1:P.

The cluster did jump in and initiated a takeover. At 21:12 (CET) the former secondary was detected as new running master (changing from 4:S to 4:P).

11.2.4.1 SAP HANA LandscapeHostConfiguration

To check the status of an SAPHana database and to find out if the cluster should react, you can use the script **landscapeHostConfiguration** to be called as Linux user *<sid>adm*.

```
suse01:~> HDBSettings.sh landscapeHostConfiguration.py
| Host   | Host   | ... NameServer   | NameServer  | IndexServer | IndexServer |
```

```
|        | Active | ... Config Role  | Actual Role | Config Role | Actual Role |
| ------ | ------ | ... ------------ | ----------- | ----------- | ----------- |
| suse01 | yes    | ... master 1     | master      | worker      | master      |

overall host status: ok
```

Following the SAP HA guideline, the SAPHana resource agent interprets the return codes in the following way:

TABLE 4: INTERPRETATION OF RETURN CODES

Return Code	Interpretation
4	SAP HANA database is up and OK. The cluster does interpret this as a correctly running database.
3	SAP HANA database is up and in status info. The cluster does interpret this as a correctly running database.
2	SAP HANA database is up and in status warning. The cluster does interpret this as a correctly running database.
1	SAP HANA database is down. If the database should be up and is not down by intention, this could trigger a takeover.
0	Internal Script Error – to be ignored.

11.3 Maintenance

To receive updates for the operating system or the SUSE Linux Enterprise High Availability Extension, it is recommended to register your systems to either a local SUSE Manager or Subscription Management Tool (SMT) or remotely with SUSE Customer Center.

11.3.1 Updating the Operating System and Cluster

For an update of SUSE Linux Enterprise Server for SAP Applications packages including cluster software, follow the rolling update procedure defined in the SUSE Linux Enterprise High Availability Extension product documentation, detailed in section *Upgrading Your Cluster and Updating Software Packages* of the SUSE Linux Enterprise High Availability Administration Guide.

11.4 Reconfiguring the Cluster After a Takeover

The nodes of the HAE Cluster monitor each other. They will shut down unresponsive or misbehaving nodes prior to any failover actions to prevent data corruption. Setting the AWS stonith-action to poweroff will permanently shut down the defect cluster node. This will expedite a takeover on AWS.

The default setting reboot makes the STONITH agent wait until a reboot has been successfully completed. This will delay the reconfiguration of the SAP HANA database. Re-integrating a faulty cluster node into the cluster needs to be performed manually since it needs investigation why the cluster node did not operate as expected.

Restarting the second (faulty) cluster node automatically can be configured as well. It bears however the risk that the remaining node gets harmed through an incorrect acting second (faulty) node. The reconfiguration of the second (faulty) node happens through the following steps:

1. Restart the node through the AWS console.
2. Investigate the node after reboot and fix a potential defect.
3. Boot SAP HANA manually. Check the instance health. Fix a potential defect. Shut down SAP HANA.
4. Configure SAP HANA to be a secondary node to the new master node.
5. Start SAP HANA as secondary node.
6. Restart the HAE cluster with the command `systemctl start pacemaker` as superuser. This process can take several minutes.
7. Verify that all cluster services operate correctly.

A takeover is now completed. The roles of the two cluster nodes have been flipped. The SAP HANA database is now protected against future failure events.

11.4.1 Updating SAP HANA - Seamless SAP HANA Maintenance

For updating SAP HANA database systems in system replication you need to follow the defined SAP processes. This section describes the steps to be done before and after the update procedure to get the system replication automated again.

SUSE has optimized the SAP HANA maintenance process in the cluster. The improved procedure only sets the master-slave-resource to maintenance and keeps the rest of the cluster (SAPHanaTopology clones and IPaddr2 vIP resource) still active. Using the updated procedure allows a seamless SAP HANA maintenance in the cluster, as the virtual IP address can automatically follow the running primary.

Prepare the cluster not to react on the maintenance work to be done on the SAP HANA database systems. Set the master-slave-resource to be unmanaged and the cluster nodes in maintenance mode.

Note

If your maintenance procedure requires a node reboot, the pacemaker service may be automatically started by `systemd` when the node comes back online. If HANA System Replication was disabled during the maintenance activities, pacemaker will **fail** to start the SAP HANA cluster resource and will throw an error message for that. This can be avoided by disabling the automatic start of the pacemaker service during boot until the maintenance is complete (`systemctl disable pacemaker`). SAP HANA System Replication must be configured and functioning normally before the pacemaker service is started and/or the cluster maintenance mode is released. We strongly recommend to follow the SAP guides on HANA update procedures.

EXAMPLE 51: MAIN SAP HANA UPDATE PROCEDURE

Pre Update Task

For the <master-slave-resource> set the maintenance mode:

```
crm resource maintenance <master-slave-resource>
```

The <master-slave-resource> in the given guide is `msl_SAPHana_HA1_HDB10`.

Update

Process the SAP Update for both SAP HANA database systems. This procedure is described by SAP.

Post Update Task

Expect the primary/secondary roles to be exchanged after the maintenance. Therefore, tell the cluster to forget about these states and to reprobe the updated SAP HANA database systems.

```
crm resource refresh <master-slave-resource>
```

After the SAP HANA update is complete on both sites, tell the cluster about the end of the maintenance process. This allows the cluster to actively control and monitor the SAP again.

```
crm resource maintenance <master-slave-resource> off
```

11.4.2 Migrating an SAP HANA Primary

In the following procedures, we assume the primary to be running on node 1 and the secondary on node 2. The goal is to "exchange" the roles of the nodes, so finally the primary should run on node 2 and the secondary should run on node 1.

There are different methods to get the exchange of the roles done. The following procedure shows how to tell the cluster to "accept" a role change via native HANA commands.

EXAMPLE 52: MIGRATING AN SAP HANA PRIMARY USING SAP TOOLSET

Pre move

Set the <master-slave-resource> to "maintenance". This could be done on any cluster node.

```
crm resource maintenance <master-slave-resource-name>
```

Manual Takeover Process

- Stop the primary SAP HANA database system. Enter the command in our example on node 1 as user *<sid>adm*.

  ```
  HDB stop
  ```

- Start the takeover process on the secondary SAP HANA database system. Enter the command in our example on node 2 as user *<sid>adm*.

  ```
  hdbnsutil -sr_takeover
  ```

- Register the former primary to become the new secondary. Enter the command in our example on node 1 as user *<sid>adm*.

  ```
  hdbnsutil -sr_register --remoteHost=suse02 --remoteInstance=10 \
  ```

```
--replicationMode=sync --name=WDF \
--operationMode=logreplay
```

- Start the new secondary SAP HANA database system. Enter the command in our example on node 1 as user *<sid>adm*.

```
HDB start
```

Post Migrate

- Wait some time until *SAPHanaSR-showAttr* shows both SAP HANA database systems to be up again (field roles must start with the digit 4). The new secondary should have role "S" (for secondary).
- Tell the cluster to forget about the former master-slave roles and to re-monitor the failed master. The command could be submitted on any cluster node as user *root*.

```
crm resource refresh master-slave-resource-name
```

- Set the <master-slave-resource> to the status managed again. The command could be submitted on any cluster node as user *root*.

```
crm resource maintenance <master-slave-resource-name> off
```

Now we explain how to use the cluster to partially automate the migration. For the described attribute query using *SAPHanaSR-showAttr* and *SAPHanaSR-filter*, you need at least SAPHanaSR with package version 0.153.

EXAMPLE 53: MOVING AN SAP HANA PRIMARY USING THE CLUSTER TOOLSET

- Create a `move` away from this node rule by using the **force** option.

```
crm resource move <master-slave-resource-name> force
```

Because of the "move away" (**force**) rule the cluster will **stop** the current primary. After that, run a **promote** on the secondary site if the system replication was in sync before. You should not migrate the primary if the status of the system replication is not in sync (SFAIL).

Important

Migration without the **force** option will cause a takeover without the former primary to be stopped. Only the migration with **force** option is supported.

Note

The `crm` resource command `move` was previously named `migrate`. The `migrate` command is still valid but already known as obsolete.

- Wait until the secondary has completely taken over to be the new primary role. You see this using the command line tool *SAPHanaSR-showAttr* and check for the attributes "roles" for the new primary. It must start with "**4:P**".

```
suse01:~ # SAPHanaSR-showAttr --format=script | \
   SAPHanaSR-filter --search='roles'
Mon Nov 11 20:38:50 2019; Hosts/suse01/roles=1:P:master1::worker:
Mon Nov 11 20:38:50 2019; Hosts/suse02/roles=4:P:master1:master:worker:master
```

- If you have set up `AUTOMATED_REGISTER="true"`, you can skip this step. In other cases you now need to register the old primary. Enter the command in our example on node 1 as user *<sid>adm*.

```
hdbnsutil -sr_register --remoteHost=suse02 --remoteInstance=10 \
    --replicationMode=sync --operationMode=logreplay \
    --name=WDF
```

- Clear the ban rules of the resource to allow the cluster to start the new secondary.

```
crm resource clear <master-slave-resource-name>
```

Note

The `crm` resource command `clear` was previously named `unmigrate`. The `unmigrate` command is still valid but already known as obsolete.

- Wait until the new secondary has started. You see this using the command line tool *SAPHanaSR-showAttr* and check for the attributes "roles" for the new primary. It must start with "**4:S**".

```
suse01:~ # SAPHanaSR-showAttr --format=script | \
   SAPHanaSR-filter --search='roles'
Mon Nov 11 20:38:50 2019; Hosts/suse01/roles=4:S:master1::worker:
Mon Nov 11 20:38:50 2019; Hosts/suse02/roles=4:P:master1:master:worker:master
```

12 Useful Links, Manuals, and SAP Notes

12.1 SUSE Best Practices and More

Blog series #towardsZeroDowntime

https://www.suse.com/c/tag/towardszerodowntime/

Best Practices for SAP on SUSE Linux Enterprise

https://documentation.suse.com/sbp/sap-12/

Blog in 2014 - Fail-Safe Operation of SAP HANA®: SUSE Extends Its High Availability Solution

http://scn.sap.com/community/hana-in-memory/blog/2014/04/04/fail-safe-operation-of-sap-hana-suse-extends-its-high-availability-solution

12.2 SUSE Product Documentation

SUSE product manuals and documentation

https://documentation.suse.com/

Current online documentation of SLES for SAP

https://documentation.suse.com/sles-sap/12-SP4/

Current online documentation of SUSE Linux Enterprise High Availability Extension

https://documentation.suse.com/sle-ha/12-SP4/

Tuning Guide for SUSE Linux Enterprise Server

https://documentation.suse.com/sles/12-SP4/html/SLES-all/book-sle-tuning.html

Storage Administration Guide for SUSE Linux Enterprise Server

https://documentation.suse.com/sles/12-SP4/single-html/SLES-storage/#stor-admin

Release Notes

https://www.suse.com/releasenotes

TID Estimate correct multipath timeout

http://www.suse.com/support/kb/doc.php?id=7008216

TID How to load the correct watchdog kernel module

http://www.suse.com/support/kb/doc.php?id=7016880

TID Addressing file system performance issues on NUMA machines

http://www.suse.com/support/kb/doc.php?id=7008919

TID Overcommit Memory in SLES

https://www.suse.com/support/kb/doc.php?id=7002775

SLES technical information

https://www.suse.com/products/server/technical-information/

XFS file system

https://www.suse.com/communities/conversations/xfs-the-file-system-of-choice/

12.3 Manual Pages

crm

crm.8

crm_simulate

crm_simulate.8

cs_clusterstate

cs_clusterstate.8

ocf_suse_SAPHana

ocf_suse_SAPHana.7

ocf_suse_SAPHanaTopology

ocf_suse_SAPHanaTopology.7

SAPHanaSR

SAPHanaSR.7

SAPHanaSR-showAttr

SAPHanaSR-showAttr.8

SAPHanaSR-replay-archive

SAPHanaSR-replay-archive.8

SAPHanaSR_manitenance_examples

SAPHanaSR_manitenance_examples.8

12.4 SAP Product Documentation

SAP HANA Installation and Update Guide

http://help.sap.com/hana/SAP_HANA_Server_Installation_Guide_en.pdf

SAP HANA Administration Guide

http://help.sap.com/hana/SAP_HANA_Administration_Guide_en.pdf

12.5 SAP Notes

1984787 - SUSE LINUX Enterprise Server 12: Installation notes

https://launchpad.support.sap.com/#/notes/1984787

2205917 - SAP HANA DB: Recommended OS settings for SLES 12 / SLES for SAP Applications 12

https://launchpad.support.sap.com/#/notes/2205917

1876398 - Network configuration for System Replication in HANA SP6

https://launchpad.support.sap.com/#/notes/1876398

611361 - Hostnames of SAP servers

https://launchpad.support.sap.com/#/notes/611361

1275776 - Preparing SLES for Sap Environments

https://launchpad.support.sap.com/#/notes/1275776

1514967 - SAP HANA: Central Note

https://launchpad.support.sap.com/#/notes/1514967

1523337 - SAP In-Memory Database 1.0: Central Note

https://launchpad.support.sap.com/#/notes/1523337

2380229 - SAP HANA Platform 2.0 - Central Note

https://launchpad.support.sap.com/#/notes/2380229

1501701 - Single Computing Unit Performance and Sizing

https://launchpad.support.sap.com/#/notes/1501701

1944799 - SAP HANA Guidelines for SLES Operating System Installation

https://launchpad.support.sap.com/#/notes/1944799

1890444 - Slow HANA system due to CPU power save mode

https://launchpad.support.sap.com/#/notes/1890444

1888072 - SAP HANA DB: Indexserver crash in strcmp sse42

https://launchpad.support.sap.com/#/notes/1888072

1846872 - "No space left on device" error reported from HANA

https://launchpad.support.sap.com/#/notes/1846872

13 Examples

13.1 Example Cluster Configuration

The following complete crm configuration is for a two-node cluster (suse01, suse02) and an SAP HANA database with SID HA1 and instance number 10. The virtual IP address in the example is 192.168.10.15.

```
node suse01
node suse02

primitive rsc_SAPHanaTopology_HA1_HDB10 ocf:suse:SAPHanaTopology \
        operations $id="rsc_sap2_HA1_HDB10-operations" \
        op monitor interval="10" timeout="600" \
        op start interval="0" timeout="600" \
        op stop interval="0" timeout="300" \
        params SID="HA1" InstanceNumber="10"
primitive rsc_SAPHana_HA1_HDB10 ocf:suse:SAPHana \
        operations $id="rsc_sap_HA1_HDB10-operations" \
        op monitor interval="61" role="Slave" timeout="700" \
        op start interval="0" timeout="3600" \
        op stop interval="0" timeout="3600" \
        op promote interval="0" timeout="3600" \
        op monitor interval="60" role="Master" timeout="700" \
        params SID="HA1" InstanceNumber="10" PREFER_SITE_TAKEOVER="true"
DUPLICATE_PRIMARY_TIMEOUT="7200" AUTOMATED_REGISTER=“false“
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 \
```

```
        meta target-role=Started \
        params tag=pacemaker profile=cluster
primitive rsc_ip_HA1_HDB10 ocf:suse:aws-vpc-move-ip \
        params ip=192.168.10.15 routing_table=rtb-XYZ interface=eth0 profile=cluster \
        op start interval=0 timeout=180 \
        op stop interval=0 timeout=180 \
        op monitor interval=60 timeout=60
ms msl_SAPHana_HA1_HDB10 rsc_SAPHana_HA1_HDB10 \
        meta clone-max="2" clone-node-max="1" interleave="true"
clone cln_SAPHanaTopology_HA1_HDB10 rsc_SAPHanaTopology_HA1_HDB10 \
        meta clone-node-max="1" interleave="true"
colocation col_saphana_ip_HA1_HDB10 2000: \
        rsc_ip_HA1_HDB10:Started msl_SAPHana_HA1_HDB10:Master
order ord_SAPHana_HA1_HDB10 2000: \
        cln_SAPHanaTopology_HA1_HDB10 msl_SAPHana_HA1_HDB10
property cib-bootstrap-options: \
        have-watchdog=false \
        dc-version=1.1.15-21.1-e174ec8 \
        cluster-infrastructure=corosync \
        stonith-enabled=true \
        stonith-action=off \
        stonith-timeout=600s \
        last-lrm-refresh=1518102942 \
        maintenance-mode=false
rsc_defaults $id="rsc_default-options" \
        resource-stickiness="1000" \
        migration-threshold="5000"
op_defaults $id="op_defaults-options" \
        timeout="600"
```

13.2 Example for /etc/corosync/corosync.conf

The following file shows a typical corosync configuration with one ring. Review the SUSE product documentation about details and about additional rings.

```
# Read the corosync.conf.5 manual page

totem {

  version: 2
  rrp_mode: passive
  token: 30000
  consensus: 36000
  token_retransmits_before_loss_const: 6
  secauth: on
```

```
  crypto_hash: sha1
  crypto_cipher: aes256
  clear_node_high_bit: yes
  interface {
    ringnumber: 0
    bindnetaddr: 10.79.254.249
    mcastport: 5405
    ttl: 1
  }

  transport: udpu

}

nodelist {
  node {
  ring0_addr: 10.79.254.249
  ring1_addr: 10.79.253.249
  nodeid: 1
  }

  node {
  ring0_addr: 10.79.9.213
  ring1_addr: 10.79.10.213
  nodeid: 2
  }
}

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
        }
}


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
}
```

13.3 Checklist - SUSE Cluster Setup in AWS

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

Checklist AWS Cluster Setup

SLES subscription and update status	
Item	**Status/Value**
All systems have a SLES for SAP subscription	
All systems have Public Cloud Module enabled	
All system have been updated to use the latest patch level	

AWS User Privileges for the installing person	
Item	**Status/Value**
Creation of EC2 instances and EBS volumes	
Creation Security Groups	
Modification of VPC routing tables	
Creation of IAM policies and attach them to IAM roles	
Potentially needed: Creation of subnets and routing tables	

VPC and Network	
Item	**Status/Value**
VPC ID	

Checklist AWS Cluster Setup	
VPC and Network	
CIDR range of VPC	
Subnet ID A for systems in AZ "A"	
Subnet ID B for systems in AZ "B"	
VPC Route table ID for Subnet A and B	
Are the VPC routing tables associated with the relevant subnets?	
Alternative: Is it associated to VPC? Subnets do not have their own ones	

AWS Policies Creation	
Item	**Status/Value**
Name of AWS Data Provider for SAP IAM policy	
Name of STONITH IAM policy	
Name of Overlay IP IAM policy	

First cluster node (initially primary server)	
Item	**Status/Value**
EC2 Instance Id	
ENI ID	
1st IP address	
2nd IP address	
Hostname	

Checklist AWS Cluster Setup	
First cluster node (initially primary server)	
Is EC2 Instance ID is associated to subnet A?	
Does the EC2 Instance has all 3 IAM policies attached?	
Is EC2 tag *pacemaker* set with hostname?	
Does the AWS CLI profile *cluster* created and set to *text*?	
Is Source/Destination Check disabled?	
Second cluster node (initially secondary server)	
Item	**Status/Value**
EC2 Instance Id	
ENI ID	
1st IP address	
2nd IP address	
Hostname	
Is the EC2 Instance is associated to subnet B?	
Does the EC2 instance has all 3 IAM policies attached?	
Is EC2 tag *pacemaker* set with hostname?	
Is AWS CLI profile *cluster* created and set to *text*?	
Is Source/Destination Check disabled?	
Overlay IP address: database service	
Item	**Status/Value**

Checklist AWS Cluster Setup	
Overlay IP address: database service	
IP address	
Has it been added to the routing tables?	
Does it point to the ENI of first node?	

Internet access	
Item	**Status/Value**
All instance have Internet access? Check routing tables	
Alternative: Add http proxies for data providers and cluster software	

14 Reference

For more detailed information, have a look at the documents listed below.

14.1 Pacemaker

Pacemaker Project Documentation

https://clusterlabs.org/pacemaker/doc/

15 Appendix: Troubleshooting

15.1 Verification and Debugging of `aws-vpc-move-ip` Resource Agent

Start the Overlay IP Address on a given Node.

With the cluster stopped or in maintenance mode, and as *root* user, run the following command using the same parameters as in your cluster configuration:

```
suse01:~ # OCF_RESKEY_address=<virtual_IPv4_address>
 OCF_RESKEY_routing_table=<AWS_route_table> OCF_RESKEY_interface=eth0
 OCF_RESKEY_profile=<AWS-profile> OCF_ROOT=/usr/lib/ocf /usr/lib/ocf/resource.d/suse/aws-
vpc-move-ip monitor
```

Check the console output (DEBUG keyword) for error messages.

Stop the Overlay IP Address on a given Node.

As *root* user run the following command using the same parameters as in your cluster configuration:

```
suse01:~ # OCF_RESKEY_address=<virtual_IPv4_address>
 OCF_RESKEY_routing_table=<AWS_route_table> OCF_RESKEY_interface=eth0
 OCF_RESKEY_profile=<AWS-profile> OCF_ROOT=/usr/lib/ocf /usr/lib/ocf/resource.d/suse/aws-
vpc-move-ip stop
```

Check the DEBUG output for errors and verify that the virtual IP address is NOT active on the current node with the command `ip address list dev eth0`. Start the overlay IP Address to be hosted on a given node.

As *root* user, run the following command using the same parameters as in your cluster configuration:

```
suse01:~ # OCF_RESKEY_address=<virtual_IPv4_address>
 OCF_RESKEY_routing_table=<AWS_route_table> OCF_RESKEY_interface=eth0
 OCF_RESKEY_profile=<AWS-profile> OCF_ROOT=/usr/lib/ocf /usr/lib/ocf/resource.d/suse/aws-
vpc-move-ip start
```

Check the DEBUG output for error messages and verify that the virtual IP address is active on the current node with the command `ip address show`.

15.2 Testing the AWS STONITH Agent

The EC2 STONITH agent will shut down the other node if he thinks that the other node stops to respond at the corosync layer. The agent can be called manually as *root* user on a cluster node 1 to shut down cluster node 2 for testing purposes.

The EC2 STONITH agent can be manually tested and validated.

Monitor Operation:

```
suse01:~ # export PATH=$PATH:/usr/share/cluster-glue
suse01:~ # stonith -t external/ec2 profile=<AWS-profile>
 tag=<aws_tag_containing_hostname> -S
```

The above command should return something like the below:

```
external/ec2[15687]: info: status check for i-abcdefg0123456789 is running
external/ec2[15677]: info: Operation status passed
info: external/ec2 device OK.
```

As part of its normal work, EC2 STONITH needs to be able to get all node's names from the EC2 resource tags. This operation can be tested as shown in the following example:

Get Nodes List Operation:

```
suse01:~ # stonith -t external/ec2 profile=<AWS-profile>
 tag=<aws_tag_containing_hostname> -l
```

The above command should return something like:

```
external/ec2[4193]: info: status check for i-abcdefg0123456789 is running
external/ec2[4183]: info: Operation gethosts passed
suse01
suse02
```

The EC2 STONITH agent should also be able to shutdown/stop the other EC2 Instance as part of a fencing operation. The fencing operation can be tested as shown in the following example:

Fencing Operation:

```
suse01:~ # stonith -t external/ec2 profile=<AWS-profile> port=<cluster-node2>
 tag=<aws_tag_containing_hostname> -T off <cluster-node2>
```

Note

The above command should shutdown/stop cluster the EC2 instance. If it does not work as expected, check the errors reported during execution of the command.

On all of the above examples the parameter used are:

- AWS-profile : The profile which will be used by the AWS CLI. heck the file ~/.aws/config for the matching one. Using the AWS CLI command aws configure list will provide the same information cluster-node2:
- The name or IP address of the other cluster node
- aws_tag_containing_hostname: The name of the tag of the EC2 instances for the two cluster nodes. We used the name *pacemaker* in this documentation

16 Legal Notice

17 GNU Free Documentation License

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all

Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
   Permission is granted to copy, distribute and/or modify this document
   under the terms of the GNU Free Documentation License, Version 1.2
```

```
   or any later version published by the Free Software Foundation;
   with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
   A copy of the license is included in the section entitled “GNU
   Free Documentation License”.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “ with…Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
   Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.