SAP HANA System Replication with SUSE HA Extension Implementation

Table of Contents

1 Background.....3
1.1 HANA High Availability Overview.....3
1.2 SAP HANA System Replication.....3
1.2.1 Simplified SAP HANA System Replication Setup.....4
1.3 SUSE High Availability Extension (HAE) Resource Agents (RA).....4
1.3.1 SAPHana Resource Agent.....5
1.3.2 SAPHanaTopology Resource Agent.....6
1.3.3 Resource Agent interface.....7
1.3.4 IP Agent.....7
1.3.5 YaST Wizard to set up SAP HANA Clusters.....7
1.4 SUSE HAE Supported Scenarios and Pre-requisites.....8
2 Installing and configuration SAP HANA System Replication.....13
2.1 Before starting.....13
2.2 SSFS Authentication.....13
2.3 Changes to SAP HANA Configuration.....13
2.4 Enable Replication.....13
2.5 Perform backup of system database and any tenants.....14
2.6 Setup HANA System Replication on Primary Node.....14
3 Installing SUSE HA Extensions and the SAP Agents.....15
3.1 Installing SLES HA Extensions.....15
3.1.1 Install SLES HA extensions and the SAP HANA Resource Agents Code.....15
3.1.2 Create the STONITH device(s).....16
3.1.3 Two Devices.....16
3.1.4 Configuration steps if using a shared device.....16
3.2 Configure SUSE HAE on primary HANA Server.....16
3.2.1 Add the Secondary HANA Server to the cluster-infrastructure.....17
3.2.2 Install the SAPHanaSR Resource Agents.....17
3.3 Configuration.....18
4 Appendix 1 SAP Notes.....20
5 Appendix 2 - Network requirements.....21
5.1 Data and log compression.....21
5.2 Replication modes.....21
5.3 Operation modes.....21
5.4 Network recommendations.....22
6 References.....24

Version details

Version	Date	Author	Description
1.0	22/08/20	Red	Initial draft for workshop

1 Background

1.1 HANA High Availability Overview

SAP HANA database runs mission critical applications and it is important that these systems remain available to users always. This requires that these systems can make faster recovery after system component failure (High Availability) or after a disaster (Disaster Recovery). This should happen without any data loss (zero RPO) and in very short recovery time (low RTO). To provide fault recovery SAP HANA software includes a watchdog function, that automatically restarts configured services (index server, name server, and so on) in case of their failure. In addition to these features, SAP and its partners offer the following high availability mechanism for SAP HANA. These solutions are based on completely redundant servers and/or storage.

- Host Auto-Failover
  One (or more) standby nodes are added to a SAP HANA system and configured to work in standby mode. In case of failure, data and log volumes of a failed worker node is taken over by a standby node. The standby node becomes a worker node and takes over user load. This solution does not need additional storage, only servers.
- SAP HANA System Replication
  SAP HANA replicates all data to a secondary SAP HANA system constantly. Data can be constantly pre-loaded in the memory of the secondary system to minimize the recovery time objective (RTO). This solution needs additional servers and storage. The focus of this reference architecture guide is SAP HANA System Replication.
- Storage Replication
  Data replication is achieved by means of storage mirroring independent from the database software. Disks are mirrored without a control process from the SAP HANA system. SAPHANA hardware partners offer this solution. This solution needs additional servers and storage.

1.2 SAP HANA System Replication

SAP HANA System Replication is implemented between two different SAP HANA systems with same number of active nodes. After system replication is setup between the two SAP HANA systems, it replicates all the data from the primary HANA system to the secondary HANA system (initial copy). After this, any logged changes in the primary system are also sent to the secondary system. The following replication modes are available for this procedure:

- Synchronous on disk (mode=sync)
  Transaction is committed after log entries are written on primary and secondary systems.
- Synchronous in memory (mode=syncmem)
  Transaction is committed after the secondary system receives the logs, but before they are written to disks.

- Asynchronous (mode=async)
  Transaction is committed after log entries are sent without any response from the secondary system.
- Full Sync
  Full synchronisation is supported by SAP but cannot be configured with SUSE HAE. Full Sync mode stops the surviving node if either node is down, so failover with SUSE HAE is not possible.

If the primary SAP HANA system fails, the system administrator must perform a manual takeover. Takeover can be performed using SAP HANA Studio or the command line. Manual failover requires continuous monitoring and could lead to longer recovery times. To automate the failover process, SUSE Linux Enterprise High Availability Extension (SUSE HAE) can be used or you can any third party vendor. The use of SUSE HAE for the takeover process helps customers achieve service level agreements for SAP HANA downtime by enabling faster recovery without any manual intervention.

1.2.1 Simplified SAP HANA System Replication Setup

SUSE Enterprise Server for SAP supports SAP HANA System Replication using components of SUSE High Availability Extension – two resource agents and a YaST wizard to simplify the cluster setup.
SUSE Enterprise Linux for SAP also includes other components to provide additional features to improve the setup of your SAP HANA environment, but are not relevant to the HA Extension:

- Malware protection with ClamSAP;
- SAP HANA Security (firewall and hardening); and
- Simplified management (Tuning with *saptune*, Storage encryption with *cryptctl* and tools to manage Corosync/Pacemaker *ClusterTools2*).

1.3 SUSE High Availability Extension (HAE) Resource Agents (RA)

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, and the slave is responsible for instances that are operated in synchronous (secondary) status.
To make configuring the cluster as simple as possible, SUSE also developed it's SAPHanaTopology resource agent. This runs on all nodes of the cluster and gathers information about the status and configuration of SAP HANA system replication. 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 Optimised;
- Cost Optimised;

- Multi-tier; and
- Multi-tenancy or MDC.

1.3.1 SAPHana Resource Agent

This resource agent from SUSE supports scale-up scenarios by checking the SAP HANA database instances for whether a takeover needs to happen. Unlike with the pure SAP solution, takeovers can be automated.

It is configured as a master/slave resource: The master assumes responsibility for the SAP HANA databases running in primary mode, whereas the slave is responsible for instances that are operated in synchronous (secondary) status. In case of a takeover, the secondary (slave resource instance) can automatically be promoted to become the new primary (master resource instance). This resource agent supports system replication for the following scale-up scenarios:

Performance-Optimised Scenario

Two servers (A and B) in the same SUSE Linux Enterprise High Availability Extension cluster, one primary (A) and one secondary (B). The SAP HANA instance from the primary server (A) is replicated synchronously to the secondary server (B).

Cost-Optimised Scenario

The basic setup of A and B is the same as in the Performance-Optimised Scenario. However, the secondary server (B) is also used for non-productive purposes, such as for an additional SAP HANA database for development or QA. T he production database is only kept on permanent memory, such as a hard disk. If a takeover needs to occur, the non-productive server will be stopped before the takeover is processed. The system resources for the productive database are then increased as quickly as possible via an SAP hook call-out script.

Chain/Multi-Tier Scenario

Three servers (A, B, and C), of which two are located in the same SUSE Linux Enterprise High Availability Extension cluster (A and B). The third server (C) is located externally. The SAP HANA system on the primary server (A) is replicated synchronously to the secondary server (B). The secondary server (B) is replicated asynchronously to the external server (C). If a takeover from A to B occurs, the connection between B and C remains untouched. However, B is not allowed to be the source for two servers (A and C), as this would be a "star" topology, which is not supported with current SAP HANA versions. Using SAP HANA commands, you can then manually decide what to do:

- The connection between B and C can be broken, so that B can connect to A.
- If replication to the external site (C) is more important than local system replication, the connection between B and C can be kept.

For all of the scenarios, SUSE Linux Enterprise Server for SAP Applications supports both single-tenant and multi-tenant (MDC) SAP HANA databases. That is, you can use SAP HANA databases that serve multiple SAP applications.

Managing the two SAP HANA database systems means that the resource agent controls the start/stop of the instances. In addition the resource agent is able to monitor the SAP HANA databases to check their availability on landscape host configuration level. For this monitoring the resource agent relies on interfaces provided by SAP. A third task of the resource agent is to also check the synchronisation status of the two SAP HANA databases. If the synchronisation is not “SOK”, then the cluster avoids a failover to the secondary side, if the primary fails. This is to improve the data consistency.

Concept of the Performance Optimised 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, we can utilise the SAP HANA resource agents included in SAPHanaSR. System Replication of the productive database is managed with SAPHana and SAPHanaTopology.

You can setup 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.

To get more insight on Scenarios, you can refer to the link provided in Reference Section.

1.3.2 SAPHanaTopology Resource Agent

To make configuring the cluster as simple as possible, SUSE has developed the SAPHanaTopology resource agent. This agent runs on all nodes of a SUSE Linux Enterprise High Availability Extension cluster and gathers information about the status and configurations of SAP HANA system replications. It is designed as a normal (stateless) clone.

This Resource Agent (RA) analyses the SAP HANA topology and “sends” all findings via the node status attributes to all nodes in the cluster. These attributes are taken by the SAPHana RA to control the SAP Hana Databases. In addition it starts and monitors the local saphostagent.

- Interface to monitor a HANA system (landscapeHostConfiguration.py):
  landscapeHostConfiguration.py has some detailed output about HANA system status and node roles. For our monitor the overall status is relevant. This overall status is reported by the returncode of the script: 0: Internal Fatal 1: ERROR 2: WARNING 3: INFO (maybe a switch the resource running) 4: OK The SAPHanaTopology resource agent will interpret returncodes 1 as NOT-RUNNING (or 1 failure) and returncodes 2+3+4 as in RUNNING. SAPHanaTopology scans the output table of landscapeHostConfiguration.py to identify the roles of the cluster node. Roles means configured and current role of the nameserver as well as the indexserver.
- Interface is hdbnsutil
  The interface hdbnsutil is used to check the “topology” of the system replication as well as the current configuration (primary/secondary) of a SAP HANA database instance. A second task of the interface is the posibility to run a system replication takeover (sr_takeover) or to register a former primary to a newer one (sr_register).

- saphostctrl
  The interface saphostctrl uses the function ListInstances to figure out the virtual host name of the SAP HANA instance. This is the hostname used during the HANA installation.

1.3.3 Resource Agent interface

The resource agent uses the following four interfaces provided by SAP:

- sapcontrol/sapstartsrv
  The interface sapcontrol/sapstartsrv is used to start/stop a HANA database instance/system
- landscapeHostConfiguration
  The interface is used to monitor a HANA system. The python script is named landscapeHostConfiguration.py. landscapeHostConfiguration.py has some detailed output about HANA system status and node roles. For our monitor the overall status is relevant. This overall status is reported by the returncode of the script: 0: Internal Fatal, 1: ERROR, 2: WARNING, 3: INFO, 4: OK The SAPHana resource agent will interpret returncodes 0 as FATAL, 1 as not-running or ERROR and and returncodes 2+3+4 as RUNNING.
- hdbnsutil
  The interface hdbnsutil is used to check the “topology” of the system replication as well as the current configuration (primary/secondary) of a SAP HANA database instance. A second task of the interface is the possibility to run a system replication takeover (sr_takeover) or to register a former primary to a newer one (sr_register).
- hdbsql / systemReplicationStatus
  Interface is SQL query into HANA (system replication table). The hdbsql query will be replaced by a python script “systemReplicationStatus.py” in SAP HANA SPS8 or 9. As long as we need to use hdbsql you need to setup secure store users for linux user root to be able to access the SAP HANA database. You need to configure a secure store user key “SAPHANASR” which can connect the SAP HANA database:
- saphostctrl
  The interface saphostctrl uses the function ListInstances to figure out the virtual host name of the SAP HANA instance. This is the hostname used during the HANA installation.

1.3.4 IP Agent

This Linux-specific resource manages IP alias IP addresses. On creating resource, virtual IP will be attached to primary site and this virtual IP will to move to secondary in case of failover.

1.3.5 YaST Wizard to set up SAP HANA Clusters

SUSE for SAP Applications now ships with a YaST wizard to manage the initial setup of clusters following the SUSE/SAP best practices. This wizard is part of the *yast2-sap-ha* package.

Note: This package is only for the initial creation of the cluster, use

```
yast2 cluster
```

to modify an existing cluster

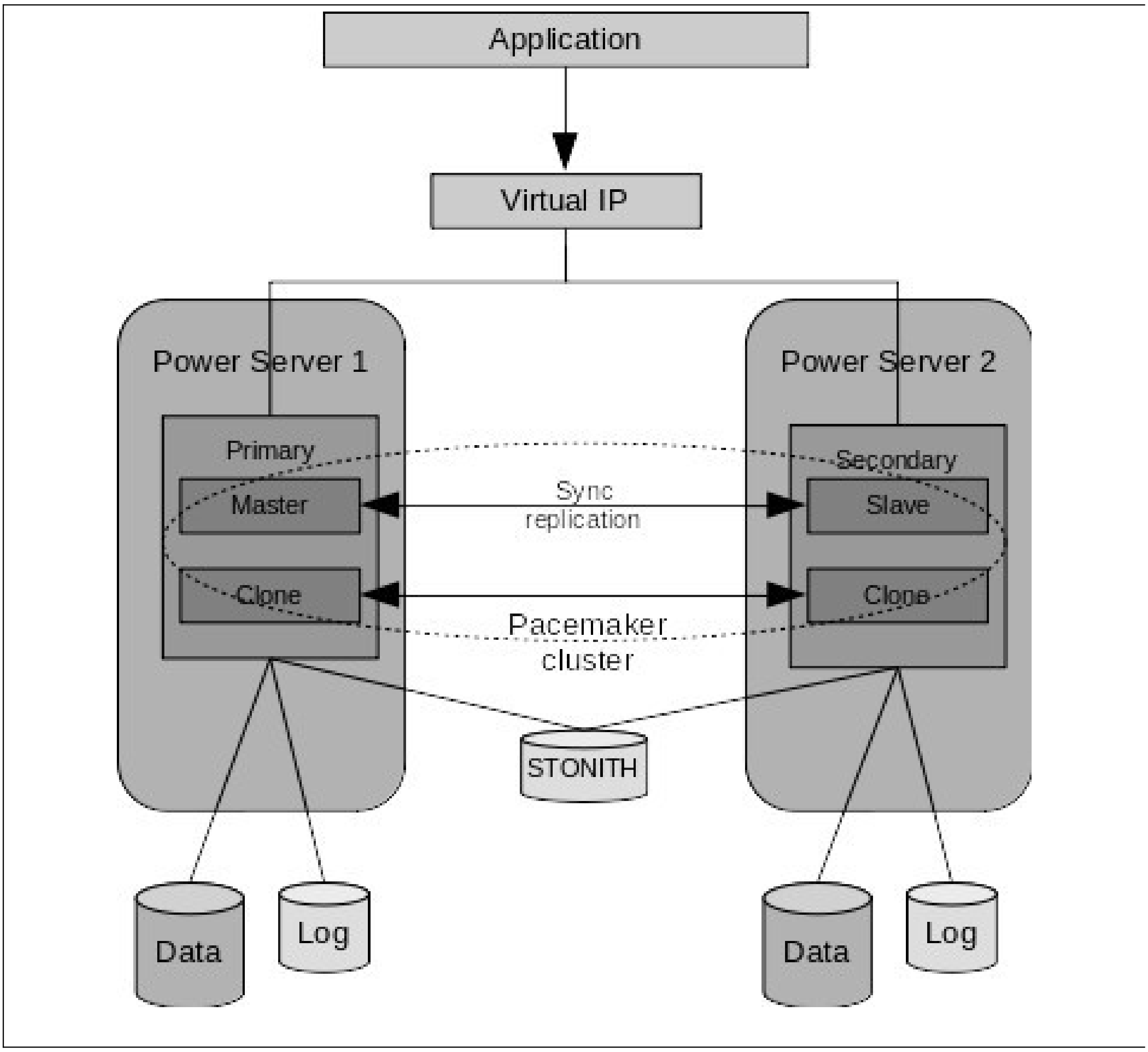
1.4 SUSE HAE Supported Scenarios and Pre-requisites

With the SAPHanaSR resource agent software package, SUSE limit the support to Scale-Up (single-box to single-box) system replication with the following configurations and parameters:

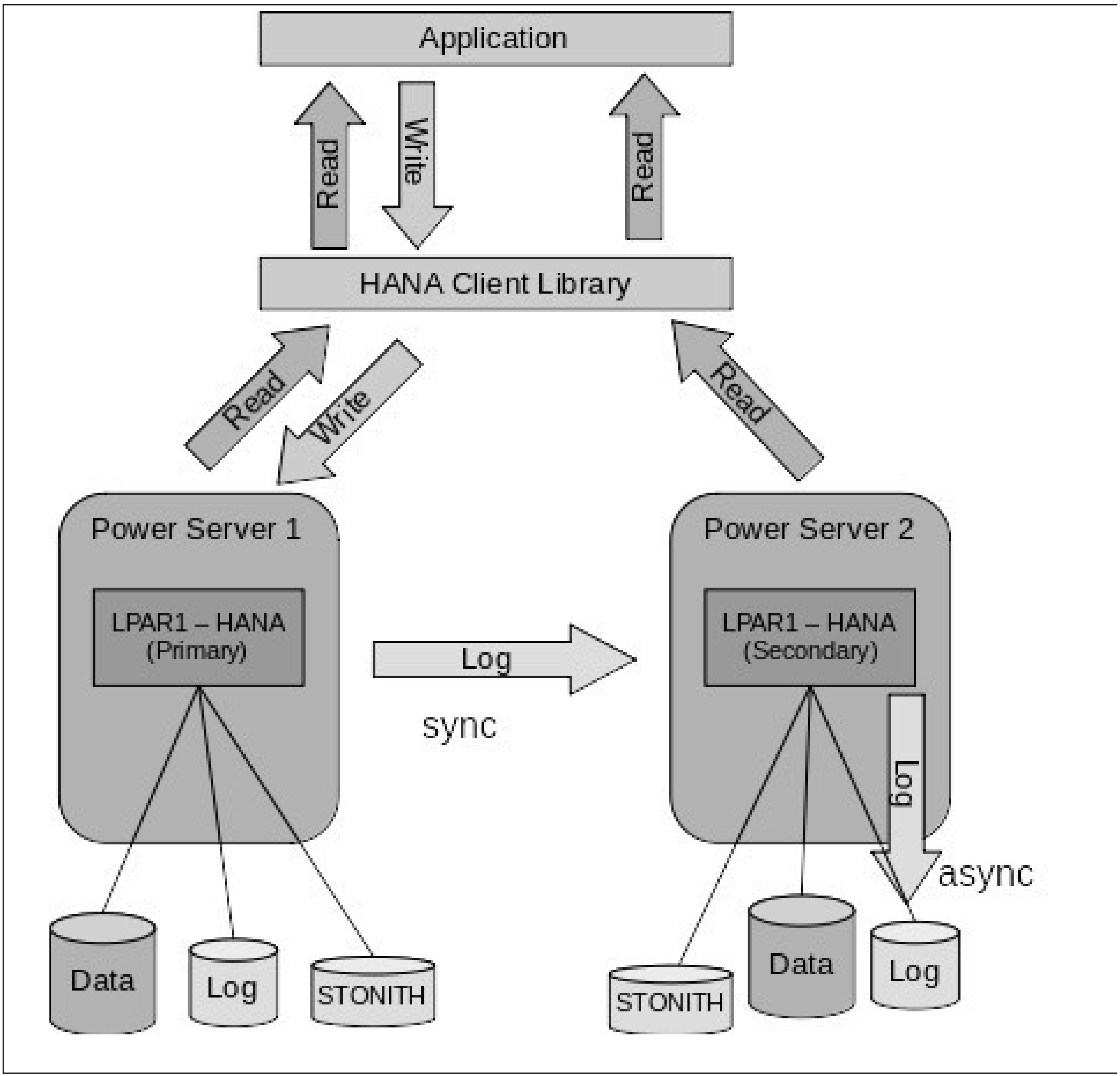
- Two-node clusters.
- The cluster must include a valid STONITH method.
    - Any STONITH mechanism supported by SLE 12 and above HAE (like SDB, IPMI) is supported with SAPHanaSR.
    - This guide is focusing on the sbd fencing method as this is hardware independent.
    - If you use SBD as the fencing mechanism, you need one or more shared drives. For productive environments, we recommend more than one SBD device.
- Both nodes are in the same network segment (layer 2).
- Technical users and groups, such as 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 synchronisation between the cluster nodes using NTP.
- Both SAP HANA instances have the same SAP Identifier (SID) and instance number.
- If the cluster nodes are installed in different data centers or data center areas, the environment must match the requirements of the SLE HAE cluster product. Of concern are the network latencies and recommended maximum distance between the nodes. Please review our product documentation for SLE HAE about those recommendations.
- Automated registration of a failed primary after takeover.
    - As a good starting configuration for projects, we recommend switching 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 hana studio or hdbnsutil.
    - For optimal automation, we recommend AUTOMATED_REGISTER=”true”.
- Automated start of SAP HANA instances during system boot must be switched off.

> Note: Valid STONITH Method – Without a valid STONITH method, the complete cluster is unsupported and will not work properly.

Solution schematic



However as the customer does not have a third site for the STONITH device, we will implement the SUSE supported option (pending SAP HANA validation)



Configuration
Parameter sheet

Parameter	Value	Description
Cluster Name		Name of cluster
Cluster Node 1		Cluster node name
Cluster Node 2		Cluster node name
SID	SLH (systemdb) SLI (tennant DB)	SAP Identifier

Instance Number		Number of the SAP HANA database. Not for system replication the instance number N+1 is blocked
Network address	w.x.y.0/nn	
Network mask	nn	
Virtual IP address 1	w.x.y.z	For DB access
Virtual IP address 2*	w.x.y.z’	(for HAWK)
Replication address node 1	a.b.c.d	Address on replication network
Replication address node 2	a.b.c.e	Address on replication network
Storage (Node 1)		Node 1 storage devices for HDB data and log
Storage (Node 2)		Node 1 storage devices for HDB data and log
SBD 1		STONITH device for Site 1 (Local or Xsite mirrored)
SBD 2		STONITH device for Site 2 (Local or Xsite mirrored)
Hawk Port	7630	
NTP Server	w.x.y.z	Address for time server (one per site??)

* One big advance of the performance optimised 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.

Requirements

SAP / SUSE

- 2 Node Cluster
- STONITH device(s)
  The Storage Based Death (SBD) device must not use host-based RAID, LVM2
- LPAR and OS configuration identical (except hostname which must be different). Any SAP or OS Configuration changes must be made on all systems. In particular users and their permissions must be the same. Automatic configuration checks will send alerts if any differences are detected,
- Both nodes on the same network segment
- Linux users defined locally
- Name resolution for all hosts and virtual address to be done locally

- The confguration of hosts in the primary and secondary systems must be the same, that is, the number of hosts must be the same but also the names of the host roles, failover groups and worker groups must be identical in both systems.
- Multiple services of the same kind (for example, index servers) on one host are not supported.
- Time synchronised between nodes
- Both SAP HANA instances have the same SAP Identifier (SID) and instance number
- Latency between the servers meets SAP requirements (See Network section) and the application requirements (that is < 1 millisecond or low single digit millisecond if application allows)
- Perform all tasks as user <sid>adm
- SAP cannot be configured to start automatically
- Version of SAP
- Version of SUSE
- Version of SUSE HA Extension

Initial requirements

- Software from SUSE: SUSE Linux Enterprise Server for SAP Applications
- installation media and a valid subscription for getting updates
- Software from SAP: SAP HANA installation media
- Physical or virtual systems including disks
- Filled parameter sheet (above)

2 Installing and configuration SAP HANA System Replication

2.1 Before starting

Confirm you have performed a data backup or storage snapshot on the primary system. In multiple-container systems, the system database and all tenant databases must be backed up. This is necessary to start creating log backups. Activated log backup is a prerequisite to get a common sync point for log shipping between the primary and secondary system.

2.2 SSFS Authentication

To prepare secondary system for authentication copy the system PKI SSFS.key and .dat files from the primary system to the secondary system. These files are found under:
/usr/sap/SID/SYS/global/security/rsecssfs/key
/usr/sap/SID/SYS/global/security/rsecssfs/data
respectivly and need to be copied to the same location on the secondary server.

2.3 Changes to SAP HANA Configuration

Changes to global.ini:
to secure communication between the Primary and Secondary servers.
[system_replication_communication]
listeninterface
allowed_sender

to ensure log segments backed up
[persistence]
log_mode

2.4 Enable Replication

On Primary server set “system_replication_hostname_resolution” to the IP address and host name of the Secondary server. On Secondary server set “system_replication_hostname_resolution” to the IP address and host name of the Primary server.

Automated registration of a failed primary after takeover.

- 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 hanastudio or hdbnsutil .

- For optimal automation, we recommend AUTOMATED_REGISTER="true" .

Automated start of SAP HANA instances during system boot must be switched off.

2.5 Perform backup of system database and any tenants

You need to perform a data backup or storage snapshot on the primary system. In multiple-container systems, the system database and all tenant databases must be backed up. This is necessary to start creating log backups. Activated log backup is a prerequisite to get a common sync point for log shipping between the primary and secondary system.

> Note: If you are creating tenant after setting up system replication, then that tenant will only become a part of system replication after performing it initial data backup.

2.6 Setup HANA System Replication on Primary Node

(see SAP Communities presentation for details, but at a high level

- On Primary System, under Configuration and Monitoring > Configure System Replication, select the option to enable system replication.
- Provide a logical name for the primary system (anything but a good idea to identify the host and data centre)
- Now stop HANA on Secondary server
- Once stopped, register the system as Secondary server – providing Logical name, Replication mode and Operation Mode
  On Primary server as user <sid>adm run:
  hdbnsutil -sr_state
  and confirm output
- Also run:
  python /sapmnt/shared/FDL/exe/linuxppc64le/hdb/python_support/systemReplicationStatus.py
  and confirm output

3 Installing SUSE HA Extensions and the SAP Agents

3.1 Installing SLES HA Extensions

Simplest method is to use the SLES bootstrap scripts. All commands from the ha-cluster-bootstrap package execute bootstrap scripts that require only a minimum of time and manual intervention.

- With ha-cluster-init , define the basic parameters needed for cluster communication. This leaves you with a running one-node cluster.
- With ha-cluster-join , add more nodes to your cluster.
- With ha-cluster-remove , remove nodes from your cluster.

All bootstrap scripts log to /var/log/ha-cluster-bootstrap.log . Check this file for any details of the bootstrap process. Any options set during the bootstrap process can be modified later with the YaST cluster module. See https://documentation.suse.com/sle-ha/15-SP1/ for details. Each script comes with a man page covering the range of functions, the script's options, and an overview of the files the script can create and modify. The bootstrap script ha-cluster-init checks and configures the following components:

NTP
If NTP has not been configured to start at boot time, a message appears.
SSH
It creates SSH keys for passwordless login between cluster nodes.
Csync2
It configures Csync2 to replicate configuration files across all nodes in a cluster.
Corosync
It configures the cluster communication system.
SBD/Watchdog
It checks if a watchdog exists and asks you whether to configure SBD as node fencing mechanism.
Virtual Floating IP
It asks you whether to configure a virtual IP address for cluster administration with Hawk2.
Firewall
It opens the ports in the firewall that are needed for cluster communication.
Cluster Name
It defines a name for the cluster, by default clusterNUMBER . This is optional and mostly useful for Geo clusters. Usually, the cluster name reflects the location and makes it easier to distinguish a site inside a Geo cluster.

3.1.1 Install SLES HA extensions and the SAP HANA Resource Agents Code

Download the media and mount on each node in turn and execute:

```
zypper install -type pattern ha_sles
```

Install the Resource Agents for controlling the SAP HANA system replication on both cluster nodes.

```
zypper in SAPHanaSR SAPHanaSR-doc
```

3.1.2 Create the STONITH device(s)

This is a method used by SUSE to avoid the problem of cluster fencing or a cluster split brain. If a member of the cluster is not behaving normally, it is removed from the cluster (Shoot The Other Node In The Head). There are multiple ways to implement STONITH, but in this configuration, STONITH Block Devices or Storage Based Death (SBD) is used and HMC device will be tested.

> Note: Currently we are unable to use a shared block device so are exploring using 2 local block devices, or an HMC

3.1.3 Two Devices

This configuration is primarily useful for environments that use host-based mirroring but where no third storage device is available. SBD will not terminate itself if it loses access to one mirror leg, allowing the cluster to continue. However, since SBD does not have enough knowledge to detect an asymmetric split of the storage, it will not fence the other side while only one mirror leg is available. Thus, it cannot automatically tolerate a second failure while one of the storage arrays is down.

Two configure, create a small (1-4MB) LUN at each site and zone to LPAR

3.1.4 Configuration steps if using a shared device

Create a small LUN (1-4 MB) on the storage array that is shared between the cluster members. Map this LUN to both primary and secondary HANA servers through storage ports. Make note of the SCSI identifier of this LUN (the SCSI identifier should be the same on both primary and secondary HANA servers). It is possible to add more than one SBD device in a cluster for redundancy. If the two HANA nodes are installed on separate storage arrays, an alternate method such as IPMI can be used for implementing STONITH.
Refer to the SUSE Linux Enterprise High Availability Extension SLE HA Guide for best practices for implementing STONITH. The validation of this reference architecture has been performed using shared storage and SBD for STONITH implementation.

3.2 Configure SUSE HAE on primary HANA Server

Run the *ha-cluster-init* command to create the cluster on the Primary Node. This step will:

- Configure for unicast or multicast
- Create ssh keys if not already created

- configure csync2 for replication of critical OS files
- Configure at least one SBD
- configure corosync ring (the second ring should be created for the replication network)
- HAWK interface (monitoring and maintenance GUI)
- A use hacluster will be created and the default password should be changed.

After making any changes, the cluster should be stopped and started again

3.2.1 Add the Secondary HANA Server to the cluster-infrastructure

The second node will be integrated into the cluster by using the *ha-cluster-join* command

> Note: If a single virtual IP was going to be used to monitor the Cluster (HAWK), it will be configured during this step.

> Note: Older versions used to set the non-quorum-policy to ignore, this is now obsolete.

3.2.2 Install the SAPHanaSR 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 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.

To configure SAPHanaSR
This is done with the Hawk wizard. Require:

- SAP SID: SAP System id (3 char string)
- SAP Instance Number: (2 digit zero filled)
- Virtual IP address: The virtual IP address used by the cluster

In Hawk GUI → Wizards → SAP → SAP HANA SR Scale-Up Performance Optimised

Enter Virtual IP address and the Virtual Host name that the SAP Application Server will use to connect to HANA Database.

Other parameters to configuration

Paramter	Performance Optimised	Cost Optimised	Multi-Tier
PREFER_SITE_TAKEOVER	True	False	True/False

AUTOMATED_REGISTER	False/True	False/True	False
DUPLICATE_PRIMARY_TIMEOUT	7200	7200	7200

PREFER_SITE_TAKEOVER	Defines whether RA should prefer to takeover the secondary instance instead of restarting the failed primary locally (may change if vPMEM)
AUTOMATED_REGISTER	Defines whether a former primary should be automatically registered to be secondary of the new primary. Used to control the level of system replication automation False: the former primary must be manually registered. The cluster will not start SAP HANA RDBMS till it is registered to avoid double primary active
DUPLICATE_PRIMARY_TIMEOUT	Time differences between two primary time stamps, if dual primary active. If the time difference is less than the time gap, then the cluster hold one or both instances in “WAITING” status. This is to give an admin the chance to react to the failover. If the complete node of the former primary crashed, the former primary will be registered only after the time difference has 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

3.3 Configuration

After configuring the above resources and constraints, the cluster will consist of the following resources

- rsc_SAPHanaTopology_<SID>_HDB<Inst#>
  This resource manages the the two SAP HANA Databases in the System Replication.
- rsc_SAPHana_<SID>_HDB<Inst#>
  analyses the SAP HANA replication topology
- rsc_ip_<SSID>_HDB
  which is a linux specific resource to manage the alias IP address (The virtual IP address which is placed as an alias on the network adapter of the Node running the Primary Database instance.
- stonith-sbd
  STONITH resource

The following two constraints will be created
Constraints are used to control resources in the cluster:

- on which cluster the resources can run
- in which order the resources are located

- what other resources a specific resource depends on

- ID: col_sapaha_ip_<SID>_HDB<Inst#>
  Score: 200
  Resources: rsc_ip_<SID>_HDB<Inst#> and msi_SAPHana_<SID>_HDB<Inst#>
- ID: ord_SAPHana_<SID>_HDB<Inst#>
  Score: 2000
  Symmetrical: Yes
  Resources: cln_SAPHanaTopology_<SID>_HDB<Inst#> and
  msl_SAPHana_<SID>_HDB<Inst#>

Background
To create a constraint, specify an ID, select the resources between which define the constraint and add a score. The score determines:

- positive values The resources should run on the same node
- negative values the resources should not run on the same node
- Infinity the resources MUST run on the same node
- -Infinity: the resources MUST not run on the same node

Ordered constraint. To create an ordered constraint, specify an ID, select the resources between which the constraint is being defined and a score (>0 the constraint is mandatory, 0 is a suggestion)

Do's and Don'ts

- Define STONITH device before adding other resources to the cluster
- Do intensive testing
- Tune the timeouts of the operations of SAPHana and SAPHanaTopology
- Start with PREFERED_SITE_TAKEOVER = true, AUTOMATED_REGISTER = false and DUPLICATE_PRIMARY_TIMEOUT = 7200

Avoid

- Rapid movement of resources between nodes
- Creating cluster without proper time synchronisation or unstable name resoulution or consistent users and groups
- Adding location rulse for the clone, master/slave or IP resource (only rules above allowed)
- As "migrating" or "moving" resources in crm-shell, HAWK, or other tools would add client prefer location rules this activity is forbidden.

4 Appendix 1 SAP Notes

SAP note	Title
2055470	HANA on Power Systems planning and installation specifics - Central note
2188482	SAP HANA on IBM Power Systems: Allowed hardware
2218464	Supported products when running HANA on Power Systems
2230704	SAP HANA on IBM Power Systems with multiple LPARs per physical host
2235581	SAP HANA: Supported operating systems
2205917	Recommended operating system settings for SUSE Linux Enterprise Server V12 and SUSE Linux Enterprise Server for SAP applications 12
1943937	Hardware Configuration Check Tool - Central note
2161344	HWCCT patch note
2460914	SAP HANA Platform V2.0 SPS 02 Release Note

5 Appendix 2 - Network requirements

Background around network requirements and testing

5.1 Data and log compression

Since SAP HANA SPS09, compression can be used to reduce the amount of traffic between the instances, an advantage especially when over long distances. It is used for the initial shipping of a full copy of the data and the subsequential delta data shipping as well as the continuous redo log shipping.

Configuration is done in the global.ini on the secondary site

```
[system_replicaton]
enable_log_compression = true
enable_data_compression = true
```

> Note: the default for both options is false, however a tail compression (appended blank characters omitted) for the log buffers is always used

5.2 Replication modes

The replication mode is only relevant for the continuous log shipping of the transactional redo log buffers, not for the actual data shipping

SAP HANA offers the following modes for shipping the transactional redo log from the Primary to Secondary sites:

- Synchronous
  The secondary system sends acknowledgement back to the Primary as soon as the data is received and write to disk acknowledged,
- Full sync
  Processing of transactions is blocked at the Primary site if the Secondary site is not available – Note this mode is not supported for HA,
- Synchronous in memory
  The secondary system sends acknowledgement back to the primary as soon as the data is received, and
- Asynchronous
  The primary does not wait for acknowledgement.

5.3 Operation modes

Prior to SAP HANA SP11, there was only the classic “delta_datashipping” mode. In SP11, a new mode was introduced - “logreplay” or “HotStandby” mode

- delta_datashipping
  In addition to the continuous redo log shipping, the secondary system requests a delta data shipping on a regular (default is every 10 minutes).
- Logreplay
  This mode is pure redo log shipping. So after the system replication was set up with a full data ship, only the redo logs are shipped. This dramatically reduces the amount of data that is transferred as we do away with the delta data ship.

5.4 Network recommendations

There are two parameters of the network that are important

- Throughput
  The throughput requirements of the network depend on the SAP HANA system replication settings
- Latency
  This is only important for the synchronous modes

Throughput
To determine the throughput requirements of the network, you will need to understand the amount of data and redo log that is generated and required to be shipped. See SAP Note 1969700 for the recommended method to determine the required network throughput.

Latency
SAP recommends that the wait time for shipping a 4KB redo log file buffers must be less than 1 millisecond (or in low single digit millisecond range).

Once you have configured SAP HANA System Replication, you can collect the log write wait times with the SQL statement "HANA_Replication_Overview – also covered in SAP Note 1969700

Network resiliency
There should be at least 2 communication channels per node, that is two or more redundant communication paths between the nodes:

- Device bonding
- A second (standby) device defined in corosync
- Redundancy in the infrastructure layer (eg the hypervisor)

It is also recommended by best practice to use a separate network for resiliency.

Network device names
The network device (card) must be the same on all nodes

Host names and addresses

- Must be configured with static addresses
- All cluster nodes must be defined in */etc/hosts* with the fully qualified host name and the short host name and nsswitch.conf configured for *hosts* as *files*

- All nodes must be able to access all other nodes via ssh. Some tools like crm_report and the Hawk2 History Explorer need passwordless access. If this does not meet regulatory requirements, there is a work around.

6 References

The following references were used:

- SAP How-to Guide: Network Required for SAP HANA System Replication
- HANA Scale-Up HA with System Replication & Automated Failover using SUSE HAE on SLES 12 SP 3 – Part 1-3 – SUSE Communities – Michael Tabron / Dennis Padia

Secondary Time Travel
https://help.sap.com/viewer/6b94445c94ae495c83a19646e7c3fd56/2.0.03/en-US/7a41aabb663e4ec793e7d344606fe616.html

Multi-Target Replication
https://help.sap.com/viewer/6b94445c94ae495c83a19646e7c3fd56/2.0.03/en-US/ba457510958241889a459e606bbcf3d3.html

Invisible Takeover
https://help.sap.com/viewer/6b94445c94ae495c83a19646e7c3fd56/2.0.03/en-US/de486b6da3074d39abbda3d7343da43a.html

SAP HANA 2.0 SPS 03 What’s New: High Availability
https://blogs.sap.com/2018/04/20/sap-hana-2.0-sps-03-whats-new-high-availability-by-the-sap-hana-academy/

SAP HANA 2.0 SPS 04 Library
https://help.sap.com/viewer/product/SAP_HANA_PLATFORM/2.0.04/en-US

Setting up pacemaker for SAP
https://blogs.sap.com/2017/11/19/be-prepared-for-using-pacemaker-cluster-for-sap-hana-part-1-basics/